

**Citrix**  
**MULTIUSER™**

*SYSTEM  
ADMINISTRATOR'S  
GUIDE*







Citrix *MULTIUSER*  
SYSTEM ADMINISTRATOR'S GUIDE



## NOTICE

Citrix Systems, Inc., ("CITRIX"), makes no representations or warranties with respect to the contents or use of this publication, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, CITRIX reserves the right to make changes in specifications and other information contained in this publication without prior notice, and without obligation to notify any person or entity of such revisions or changes.

© Copyright 1990 by Citrix Systems, Inc. All rights reserved.

© Copyright Microsoft Corporation, 1981-1990. All rights reserved.

Patent #4,825,358

Patent #4,779,187

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's personal use, without the express prior written permission of:

Citrix Systems, Inc.  
210 University Drive, Suite 700  
Coral Springs, FL 33071

Printed in the United States of America.

## TRADEMARKS

The following names, used in this publication, are trademarks or registered trademarks:

Citrix and Citrix <i>MULTIUSER</i>	Citrix Systems, Inc.
Microsoft	Microsoft Corporation
MS	Microsoft Corporation
MS DOS	Microsoft Corporation
Codeview	Microsoft Corporation
MS OS/2	Microsoft Corporation
Microsoft Excel	Microsoft Corporation
Microsoft Word	Microsoft Corporation
IBM, PS/2, and OS/2	International Business Machines Corporation
IBM Quietwriter III	International Business Machines Corporation
PC/AT	International Business Machines Corporation
Micro Channel	International Business Machines Corporation
Presentation Manager	International Business Machines Corporation
Compaq	Compaq Computer Corporation
Hayes	Hayes Microcomputer Products Incorporated
386	Intel Corporation
National Semiconductor 16450	National Semiconductor Corporation
Esprit Opus 3n1+	Esprit Systems Incorporated
Kimtron KT-70PC	Kimtron Corporation
MC2, LINK MC2PC, and LINK MC5	Link Technologies Incorporated
IBM 3151	International Business Machines Corporation
ADDS 2025	Applied Digital Data Systems Incorporated
Wyse	Wyse Technology
Wyse 60 and Wyse 150	Wyse Technology
WY-60 and WY-150	Wyse Technology
Okidata and Okitel	Oki Electric Industry Company
Okidata Okitel 2400	Oki Electric Industry Company
Multitech and Multitech 224E	Multi-Tech Systems Incorporated
Televideo	Televideo Systems Incorporated
Televideo 965	Televideo Systems Incorporated
Boca Research IO/AT	Boca Research Incorporated
Digiboard and Digiboard PC/Xi	Digiboard Incorporated
Digiboard MC/Xi	Digiboard Incorporated
AST and AST I/O Mini II	AST Research Incorporated

## TABLE OF CONTENTS

WELCOME TO Citrix <i>MULTIUSER</i> .....	1
OVERVIEW .....	1
Role of the System Administrator .....	2
HOW TO USE THIS GUIDE .....	3
Notational Conventions .....	5
Key Combinations .....	6
Syntax Conventions .....	7
 PART 1: SYSTEM OVERVIEW .....	 1-1
CHAPTER 1: INTRODUCTION .....	1-1
OVERVIEW .....	1-2
USERS AND GROUPS .....	1-2
LOGINS AND SESSIONS .....	1-4
SECURITY .....	1-9
EVENT LOGGING .....	1-12
RESOURCE MANAGEMENT .....	1-13
Maximum Resource Limit .....	1-15
Minimum Resource Limit .....	1-15
Resources Managed .....	1-16
System Global/User Login Global Shared Resources .....	1-16
PRINTING AND SPOOLING .....	1-17
TERMINALS .....	1-19
THE HIGH-PERFORMANCE FILE SYSTEM (HPFS) .....	1-20
SYSTEM USER/TERMINAL STARTUP OPTIONS .....	1-21
 PART 2: SYSTEM INSTALLATION .....	 2-1
CHAPTER 2: SYSTEM INSTALLATION .....	2-1
INTRODUCTION .....	2-2
BEFORE YOU INSTALL MS OS/2 <i>MULTIUSER</i> .....	2-2
INSTALLING FILE SYSTEMS .....	2-3
Installing with Multiple Partitions .....	2-4
Installing with Multiple Fixed Disks .....	2-6
RUNNING THE INSTALLATION PROGRAM .....	2-6
Getting Help .....	2-7

AFTER MS OS/2 <i>MULTIUSER</i> INSTALLATION IS COMPLETE .....	2-7
Starting MS OS/2 <i>MULTIUSER</i> .....	2-7
CHAPTER 3: SETTING UP YOUR FIXED DISK PARTITIONS .....	3-1
INTRODUCTION .....	3-2
CONFIGURING A FIXED DISK .....	3-2
Partitions and Logical Drives .....	3-3
Setting Up Your Fixed Disk .....	3-4
Starting the FDISK Program .....	3-5
Creating the Primary Partition .....	3-7
Creating an Extended Partition .....	3-9
Creating Logical Drives on the Extended Partitions .....	3-10
Deleting a Partition .....	3-13
Deleting a Logical Drive .....	3-14
Deleting the Extended Partition .....	3-15
Deleting the Primary Partition .....	3-16
Displaying Partition Data .....	3-16
Selecting the Next Fixed Disk Drive .....	3-18
PART 3: SYSTEM CONFIGURATION .....	4-1
CHAPTER 4: SYSTEM CONFIGURATION OVERVIEW .....	4-1
INTRODUCTION .....	4-2
SETTING UP THE FIRST USER PROGRAM .....	4-2
PROGRAM SELECTOR VS COMMAND VS APPLICATION .....	4-2
USING STARTUP FILES .....	4-3
Configuration Files .....	4-6
Batch Files .....	4-7
MULTIUSER DIRECTORY STRUCTURE .....	4-7
INTRODUCTION TO THE CONFIG UTILITY .....	4-8



CHAPTER 5: CONFIGURING PROFILES .....	5-1
INTRODUCTION .....	5-2
PROFILE NAMES .....	5-3
RESOURCE MANAGEMENT .....	5-3
Maximum Resource Limit .....	5-4
Minimum Resource Limit .....	5-4
SYSTEM PROFILE .....	5-5
GROUP PROFILES .....	5-11
Creating a New Group .....	5-13
Copying a Group .....	5-14
Deleting a Group .....	5-15
Modifying a Group .....	5-15
Renaming a Group .....	5-17
USER PROFILES .....	5-18
Creating a New User .....	5-21
Copying a User .....	5-22
Deleting a User .....	5-23
Modifying a User .....	5-24
Renaming a User .....	5-25
Adding Users to a Group .....	5-26
Deleting a User From a Group .....	5-27
Changing a User's Password .....	5-28
 CHAPTER 6: CONFIGURING TERMINAL SUBSYSTEMS AND TERMINALS .....	6-1
TERMINOLOGY .....	6-2
BACKGROUND ON ADDING A TERMINAL SUBSYSTEM .....	6-4
TERMINAL SUBSYSTEMS .....	6-5
Installing an Adapter .....	6-5
Hardware Installation .....	6-5
Software Installation .....	6-5
Adding A Terminal Subsystem .....	6-6
Deleting a Terminal Subsystem .....	6-8
CONFIGURING TERMINALS, CABLES, AND SERIAL PORTS .....	6-9
Setting up Your Terminal .....	6-9
Creating a New Terminal Profile .....	6-10
Modifying a Terminal Profile .....	6-13

Terminal Parameters to Modify .....	6-14
Terminal Type .....	6-14
Autologin Parameters .....	6-14
Initial Terminal State .....	6-15
Hotkeys .....	6-16
Host to Serial Terminal Communications .....	6-18
Data Flow .....	6-19
Data Format and Handshaking Techniques .....	6-20
Data Bits .....	6-20
Parity .....	6-21
Stop Bits .....	6-21
Flow Control .....	6-21
Connection Type .....	6-23
Receive Buffer Size .....	6-25
Configuring a Dial-In Terminal .....	6-25
Modem Hardware Requirements .....	6-26
Deleting a Terminal Profile .....	6-27
Renaming a Terminal Profile .....	6-27
Moving a Terminal Profile .....	6-28
Copying a Terminal Profile .....	6-29
Changing a Terminal Type .....	6-30
Changing Device Name .....	6-31
Using System Defaults .....	6-31
Using Terminal Type Defaults .....	6-32
Resetting Terminal Configuration Parameters .....	6-32
Refreshing Configuration Parameters .....	6-33
Creating a New Terminal Subsystem .....	6-33
Reserving a Serial Port for Custom Use .....	6-34
 CHAPTER 7: CONFIGURING SECURITY .....	 7-1
USERS AND GROUPS .....	7-2
LOGIN .....	7-4
Passwords .....	7-4
Autologin .....	7-5



SECURITY CLASSES .....	7-5
Administrator .....	7-6
Operator .....	7-7
User .....	7-8
Guest .....	7-8
SECURITY ATTRIBUTES .....	7-9
Access Control List (ACL) .....	7-9
Audit Attributes .....	7-11
Ownership .....	7-11
SYSTEM RESOURCE SECURITY .....	7-12
File System Security .....	7-12
Classes and File System Access .....	7-12
Device Security and Serialization .....	7-13
Printer Security .....	7-14
Terminal Security .....	7-14
User Functions .....	7-15
Program Interfaces and Functions .....	7-15
REGISTERED PROGRAMS .....	7-16
Securing the MS OS/2 <i>MULTIUSER</i> Programming Interface .....	7-16
Securing MS OS/2 <i>MULTIUSER</i> Program Functions .....	7-17
Access Permissions for Programs .....	7-17
SECURITY AND PROGRAM LOADING .....	7-18
Security Checks Made During the Load Process .....	7-19
What Happens When a Load Fails .....	7-19
How to Correct a Loading Problem .....	7-20
RESTRICTIONS ON CHANGING PERMISSIONS .....	7-21
SECURITY PERMISSIONS .....	7-22
File System Permissions .....	7-22
Device Permissions .....	7-23
Terminal Permissions .....	7-24
API Permissions .....	7-24
Program Function Point Permissions .....	7-24
User Function Point Permissions .....	7-24
DISABLING SECURITY ACCESS CHECKS .....	7-25
SECURITY AUDITING .....	7-26
Audit Events .....	7-26
Audit Targets .....	7-27

Audit Log .....	7-28
Secondary Audit Events .....	7-28
SECURITY ATTRIBUTES AND INSTALLATION .....	7-28
Resource Security Layout .....	7-29
What Happens to the \USR\xxx Directory .....	7-30
PRESERVING SECURITY ATTRIBUTES .....	7-30
USING THE CONFIG ACCESS UTILITY .....	7-32
Access Name .....	7-33
Expanded and Compressed Directories .....	7-33
Permissions .....	7-34
Explicit Versus Implicit ACL .....	7-34
Audit Security Attribute Indicator .....	7-34
Viewing Files Within a Directory .....	7-34
Refreshing the Menu (F5) .....	7-35
Viewing the Configuration Options .....	7-35
Changing the Access Name .....	7-36
Changing the Permissions for Multiple Directories or Files .....	7-37
Viewing an ACL .....	7-38
Viewing the Audit Security Attribute (ASA) .....	7-41
Removing the Audit Security Attribute .....	7-42
Configuring Other Drives .....	7-42
Expanding All Subdirectories .....	7-42
Changing the Path .....	7-43
Configuring Security Attributes for Resources Other Than Files and Directories .....	7-43
CHAPTER 8: CONFIGURING PRINTING AND SPOOLING .....	8-1
INTRODUCTION .....	8-2
PRINT QUEUES .....	8-2
Adding a Print Queue .....	8-2
Changing a Print Queue .....	8-5
Deleting a Print Queue .....	8-6
CONFIGURE A COMMUNICATION PORT .....	8-6
CONFIGURE A PRINTER PORT .....	8-8
SPOOL OPTIONS .....	8-9
Enable/Disable Spooling .....	8-9
Setting the Spool Path .....	8-10
Setting the Refresh Rate .....	8-11

REDIRECTING A PRINTER PORT .....	8-12
CREATING A SEPARATOR PAGE FILE .....	8-13
CHAPTER 9: USING SYSTEM CONFIGURATION ( <i>CONFIG.SYS</i> ) COMMANDS .	9-1
INTRODUCTION .....	9-2
DEFINING THE <i>CONFIG.SYS</i> FILE .....	9-3
MODIFYING VALUES OF CONFIGURATION COMMANDS .....	9-5
ENABLING INPUT/OUTPUT PRIVILEGE LEVELS .....	9-6
MODIFYING MULTITASKING FEATURES .....	9-7
Specifying the Maximum Number of Threads .....	9-8
Specifying the Maximum Time a Process Waits .....	9-9
Disabling Dynamic Priority Assignments .....	9-10
Setting the Time-Slice Values .....	9-10
MANAGING SYSTEM MEMORY .....	9-11
Setting Up a Disk Cache .....	9-11
Enabling Memory Swapping and Moving .....	9-12
Using a Disk-Swap File .....	9-13
Changing the Number of Disk Buffers .....	9-14
SETTING UP A DEVICE DRIVER .....	9-14
SETTING UP YOUR ENVIRONMENT VARIABLES .....	9-15
Predefined User Environment Variables .....	9-16
Inheriting Environment Variables .....	9-16
STARTING A PROGRAM WHEN YOU START YOUR SYSTEM .....	9-17
SETTING UP CODE-PAGE SUPPORT .....	9-18
Setting the Country Code .....	9-19
Preparing Code Pages .....	9-20
Preparing Devices for Use with Code Pages .....	9-20
Keyboard .....	9-20
Screen .....	9-21
Printer .....	9-22
System Console Code Page Support .....	9-23
Terminal Code Page Support .....	9-23
CONTROLLING SYSTEM TRACING .....	9-23
Setting the Size of the System-Trace Buffer .....	9-24
Turning Tracing On or Off .....	9-24
<i>CONFIG.SYS</i> .....	9-24
TRACE Utility .....	9-25
CONTROLLING SYSTEM LOGGING .....	9-25



ENABLING AND DISABLING SECURITY ACCESS CHECKS .....	9-26
ENABLING AND DISABLING RESOURCE MANAGEMENT .....	9-26
CONTROLLING CONSOLE KEYBOARD HOTKEYS .....	9-27
CHAPTER 10: USING Citrix <i>MULTIUSER</i> DEVICE DRIVERS .....	10-1
INTRODUCTION .....	10-2
SETTING UP AN MS OS/2 <i>MULTIUSER</i> DEVICE DRIVER .....	10-3
CHANGING A DISPLAY DEVICE DRIVER .....	10-4
USING A SERIAL COMMUNICATIONS PORT .....	10-6
USING A LOGICAL DRIVE LETTER .....	10-7
USING A VIRTUAL DISK DRIVE .....	10-8
CHAPTER 11: CONFIGURING APPLICATIONS .....	11-1
INTRODUCTION .....	11-2
DIRECTORY STRUCTURE AND SECURITY CONSIDERATIONS .....	11-2
Categorizing Applications .....	11-3
When Installing an Application .....	11-4
<i>CONFIG.SYS</i> and Security .....	11-5
Setting up Users for the Installed Application .....	11-6
Access to the Programs .....	11-7
Access to the Data Paths .....	11-7
SYSTEM GLOBAL/USER LOGIN GLOBAL CONSIDERATIONS	
FOR APPLICATIONS .....	11-9
When to Use the REGISTER Utility .....	11-10
How to Use the REGISTER Utility .....	11-10
TIME CRITICAL PRIORITY CONSIDERATIONS .....	11-11
When to REGISTER a Program for SET_TIME_CRITICAL Access .....	11-12
How to REGISTER a Program for SET_TIME_CRITICAL Access .....	11-12
PART 4: SYSTEM OPERATION AND MAINTENANCE .....	12-1
CHAPTER 12: SYSTEM OPERATION .....	12-1
HANDLING LOGIN AND LOGOUT .....	12-2
Introduction .....	12-2
Logging In As The Administrator .....	12-2
Login Example .....	12-3
Changing the Password Example .....	12-4
Logging Out .....	12-4

Logout Examples .....	12-4
Handling User Logins .....	12-5
Managing the Licensed Number of Users .....	12-5
Autologin for Users .....	12-6
Handling User Passwords .....	12-7
Logging Users Out .....	12-7
User Logout Examples .....	12-7
SHUTTING DOWN THE SYSTEM .....	12-8
USING FILES AND DIRECTORIES IN HPFS .....	12-9
Naming HPFS Files and Directories .....	12-9
Using HPFS File and Directory Names with Commands .....	12-10
Transferring HPFS Files and Directories .....	12-10
Using Wildcard Characters in HPFS .....	12-11
BACKING UP AND RESTORING FILES AND DIRECTORIES .....	12-12
Making Backup Copies of Your Files .....	12-12
Making Backup Files on Another Disk .....	12-13
Making Backup Copies of Subdirectories .....	12-15
Making Backup Copies of Files Modified Since the Last Backup Operation .....	12-15
Adding Files to the Backup Disk .....	12-16
Making Backup Copies of Files Modified After a Certain Date and Time .....	12-16
Making a Backup Log .....	12-17
Preserving Extended Attributes .....	12-17
Preserving Security Attributes .....	12-17
Restoring Backup Files .....	12-18
Restoring Files to a Disk .....	12-18
Restoring Subdirectories .....	12-18
Prompting the User Before Restoring Files .....	12-19
Selecting Files to Restore .....	12-19
Restoring Files That Are Not on the Destination Drive .....	12-20
Restoring Extended Attributes and Security Attributes .....	12-20
WORKING WITH FIXED AND FLOPPY DISKS .....	12-21
Formatting a Disk .....	12-21
Formatting a Floppy Disk .....	12-22
Formatting a 5¼-Inch Disk .....	12-23
Formatting a 3½-Inch Disk .....	12-24
Copying a Floppy Disk .....	12-25

Displaying the Volume Identification .....	12-26
Assigning or Changing the Volume Label .....	12-27
Comparing Floppy Disks .....	12-28
QUERYING SYSTEM STATUS .....	12-29
Introduction .....	12-29
QUERY AUDIT .....	12-30
QUERY EVENTS .....	12-30
QUERY HOST .....	12-31
QUERY LIMITS .....	12-32
QUERY LOGIN .....	12-34
QUERY MEMORY .....	12-35
QUERY PRINT .....	12-38
QUERY PROCESS .....	12-38
QUERY SESSION .....	12-40
QUERY TERMINAL .....	12-42
QUERY USER .....	12-43
CHANGING SETTINGS .....	12-45
Introduction .....	12-45
Audit .....	12-45
Events .....	12-46
User Limits .....	12-46
Terminal .....	12-47
RESETTING SETTINGS .....	12-48
Introduction .....	12-48
Audit .....	12-48
Events .....	12-49
User Limits .....	12-49
Terminal .....	12-50
TERMINATING A PROCESS .....	12-51
SENDING MESSAGES TO USERS .....	12-51
SETTING DATE AND TIME .....	12-52
Setting and Displaying the Date .....	12-52
Setting and Displaying the Time .....	12-53
SPECIFYING DEVICE INFORMATION (MODE Utility) .....	12-54
Introduction .....	12-54
Setting the Parallel Printer Mode .....	12-55
Setting the Asynchronous Communication Mode for Custom Async Devices .....	12-56
Setting the Display Mode .....	12-59



CHAPTER 13: SYSTEM MAINTENANCE .....	13-1
MAINTENANCE MODE .....	13-2
Introduction .....	13-2
How To Enter And Exit Maintenance Mode .....	13-3
DISK MAINTENANCE .....	13-4
Introduction .....	13-4
Displaying Disk Information .....	13-5
Fixing Errors on Your Disk .....	13-7
Displaying Each File on Your Disk .....	13-9
Recovering a File From a Damaged HPFS Disk .....	13-10
Recovering a File From a Damaged FAT Disk .....	13-11
BACKING UP AND RESTORING THE SYSTEM CONFIGURATION .....	13-12
Introduction .....	13-12
Critical System Directories .....	13-13
How to Backup the System Profiles .....	13-14
Restoring the System .....	13-15
BACKING UP SYSTEM FILES THAT ARE OPEN .....	13-16
CHAPTER 14: SYSTEM EVENT LOGGING .....	14-1
INTRODUCTION .....	14-2
GENERAL SYSTEM LOGGING FACILITY .....	14-2
ENABLING AND DISABLING THE SYSTEM LOG FACILITY .....	14-2
MANAGING THE LOG FILES .....	14-3
Naming the Log File .....	14-3
SELECTING EVENTS TO LOG .....	14-4
FORMATTING THE LOG DATA .....	14-5
A CAUTION ABOUT SECURITY AUDITING .....	14-6
CHAPTER 15: SPOOLER OPERATION .....	15-1
INTRODUCTION .....	15-2
SPOOL UTILITY PANEL .....	15-2
Refresh the SPOOL Utility Panel .....	15-3
Detail Zoom the SPOOL Utility Panel .....	15-3
PRINT QUEUE CHANGES .....	15-3
Hold a Print Queue .....	15-3
Release a Print Queue .....	15-4
Cancel All Jobs in a Print Queue .....	15-4

PRINT JOB CHANGES .....	15-4
Cancel a Job .....	15-4
Print Job Next .....	15-5
Start the Job Again .....	15-5
Hold the Job .....	15-6
Release a Job .....	15-6
CHAPTER 16: PROBLEM DETERMINATION .....	16-1
TERMINAL DOESN'T WORK .....	16-2
CANNOT ACCESS DISK BECAUSE OF SECURITY .....	16-4
FORGOTTEN PASSWORD .....	16-6
LAST ADMINISTRATOR DELETED .....	16-7
CANNOT DELETE A FIXED DISK PARTITION .....	16-8
USERS FAILING DUE TO LACK OF SYSTEM RESOURCES .....	16-8
Users Cannot Login Due to Lack of Resources .....	16-8
Out of Resource Hard Error Popups .....	16-9
User Maximum Limit Reached .....	16-9
Request Would Interfere With Other User's Minimum Guaranteed Limits .....	16-10
AN APPLICATION DOES NOT RUN PROPERLY .....	16-11
Security Considerations .....	16-11
Popup Indicating Error Code SYS0005 .....	16-11
Access Denied During Load .....	16-11
Access Denied During Application Execution .....	16-11
System Global/User Login Global Considerations .....	16-11
Time Critical Priority Considerations .....	16-12
APPENDIX A: <i>README</i> FILES .....	A-1
WHAT IS A <i>README</i> FILE .....	A-1
HOW TO READ OR PRINT A <i>README</i> FILE .....	A-1
APPENDIX B: DIFFERENT WAYS TO SET UP USERS AND SYSTEM STARTUP ..	B-1
SCENARIO 1 .....	B-1
SCENARIO 2 .....	B-2
SCENARIO 3 .....	B-3
SCENARIO 4 .....	B-4



APPENDIX C: INSTALLING ASYNC PORTS FOR Citrix <i>MULTIUSER</i> . . . . .	C-1
GENERAL CONSIDERATIONS . . . . .	C-1
UNINTELLIGENT ASYNC CARD SETUP . . . . .	C-2
Boca Resarch IO/AT Board . . . . .	C-2
AST I/O Mini II Card . . . . .	C-4
INTELLIGENT ASYNC CARD SETUP . . . . .	C-5
DigiBoard PC/Xi . . . . .	C-5
DigiBoard MC/Xi . . . . .	C-6
APPENDIX D: SETTING UP SPECIFIC TERMINAL TYPES . . . . .	D-1
SOME GENERAL CONSIDERATIONS . . . . .	D-1
ADDS 2025 Terminal . . . . .	D-3
Esprit OPUS 3nl+ Terminal . . . . .	D-5
IBM 3151 Terminal (Models 51/61) . . . . .	D-8
Kimtron KT-70PC Terminal . . . . .	D-10
Link MC2PC Terminal . . . . .	D-11
Link MC5 Terminal . . . . .	D-13
Televideo 965 Terminal . . . . .	D-16
Wyse 60 Terminal . . . . .	D-18
Wyse 150 Terminal . . . . .	D-21
APPENDIX E: CABLE TYPES AND CONFIGURATION IMPLICATIONS . . . . .	E-1
RS-232 STANDARD DTE AND DCE PIN DEFINITIONS . . . . .	E-2
APPENDIX F: SETTING UP SPECIFIC MODEM TYPES . . . . .	F-1
HARDWARE REQUIREMENTS . . . . .	F-1
MODEM SWITCH SETTINGS . . . . .	F-2
MultiTech 224E Modem . . . . .	F-3
Okidata Okitel 2400 Modem . . . . .	F-3
SOFTWARE PORT CONFIGURATION . . . . .	F-3
GLOSSARY . . . . .	Glossary-1
INDEX . . . . .	I-1



## CHAPTER QUICK REFERENCE

WELCOME TO Citrix *MULTIUSER*

CHAPTER 1: INTRODUCTION

CHAPTER 2: SYSTEM INSTALLATION

CHAPTER 3: SETTING UP YOUR FIXED DISK PARTITIONS

CHAPTER 4: SYSTEM CONFIGURATION OVERVIEW

CHAPTER 5: CONFIGURING PROFILES

CHAPTER 6: CONFIGURING TERMINAL SUBSYSTEMS AND TERMINALS

CHAPTER 7: CONFIGURING SECURITY

CHAPTER 8: CONFIGURING PRINTING AND SPOOLING

CHAPTER 9: USING SYSTEM CONFIGURATION (*CONFIG.SYS*) COMMANDS

CHAPTER 10: USING Citrix *MULTIUSER* DEVICE DRIVERS

CHAPTER 11: CONFIGURING APPLICATIONS

CHAPTER 12: SYSTEM OPERATION

CHAPTER 13: SYSTEM MAINTENANCE

CHAPTER 14: SYSTEM EVENT LOGGING

CHAPTER 15: SPOOLER OPERATION



## CHAPTER QUICK REFERENCE

CHAPTER 16: PROBLEM DETERMINATION

APPENDIX A: *README* FILES

APPENDIX B: DIFFERENT WAYS TO SET UP USERS AND SYSTEM STARTUP

APPENDIX C: INSTALLING ASYNC PORTS FOR Citrix *MULTIUSER*

APPENDIX D: SETTING UP SPECIFIC TERMINAL TYPES

APPENDIX E: CABLE TYPES AND CONFIGURATION IMPLICATIONS

APPENDIX F: SETTING UP SPECIFIC MODEM TYPES

GLOSSARY



# WELCOME TO Citrix *MULTIUSER*

## OVERVIEW

Multiuser systems require some degree of administration and operation in order to distribute the system resources properly and fairly, and to control some degree of separation of users. Administration is simply a functional category of the system and does not imply or require a certain user or user set to invoke these functions. In fact, all users participate to some degree in system administration; a case in point is the login/logout sequence.

Because Citrix *MULTIUSER* provides multiuser extensions to MS OS/2, it will be referred to in this publication as MS OS/2 *MULTIUSER*.

This guide discusses several aspects of the administration and operation of MS OS/2 *MULTIUSER*.

The goal of MS OS/2 *MULTIUSER* administration facilities is twofold:

1. Provide a simple environment for smaller, less sophisticated system installations.
2. Provide a robust, highly configurable set of features for more complex system installations.

Consequently, MS OS/2 *MULTIUSER* has the following characteristics:

- It can support a completely open system, where all users have complete access in a "free-for-all" environment; however, this is not a characteristic of the base system.

- It can support a "standard" configuration providing a comfortable mixture of ease-of-use with logical separation of user resources. Separation of user capabilities is provided along security class lines. There can be people in the following four security classes: Administrator, Operator, User, and Guest.

The system design of all components of MS OS/2 *MULTIUSER* provide the mechanisms to support the variety of environments. The administration tools are used to describe the desired policies of the installation.

## Role of the System Administrator

The System Administrator fills many major roles:

- Installs MS OS/2 *MULTIUSER* on appropriate hardware.
- Configures users and how they start up their use of the system.
- Configures, sets up, and connects terminals.
- Configures security.
- Configures resource management.
- Installs applications.
- Handles user problems and questions.

The System Administrator can be any user or set of users who are in the Administrator security class. In addition to system administration, the operation and maintenance of MS OS/2 *MULTIUSER* is also discussed in this book.



## HOW TO USE THIS GUIDE

This guide is organized as an easy to use reference and is arranged by task. It is not necessary to read it straight through. You are given instructions for accomplishing each task; some shortcuts and alternate methods are also provided.

This guide is divided into a Welcome chapter, four parts, Appendices, a Glossary, and an Index. It is designed for System Administrators who are capable of performing tasks more complicated than is necessary for the typical MS OS/2 *MULTIUSER* user.

In this guide, after the Welcome chapter you will find the following Parts:

### Part 1: Introduction

Gives an overview of the MS OS/2 *MULTIUSER* system; some of the major features are highlighted.

### Part 2: System Installation

Chapters in this Part explain basic concepts the System Administrator needs when installing MS OS/2 *MULTIUSER*, such as:

- Considerations before installing the system.
- Running the installation program.
- Setting up fixed disk partitions.

## Part 3: System Configuration

Chapters in this Part describe how the System Administrator configures and customizes the MS OS/2 *MULTIUSER* system. The following topics are discussed:

- Configuring the system, group, and user profiles.
- Configuring terminals.
- Configuring security.
- Configuring printing and spooling.
- Changing *CONFIG.SYS* file.
- Using MS OS/2 *MULTIUSER* device drivers.
- Configuring applications.

## Part 4: System Operation and Maintenance

Chapters in this Part describe how the System Administrator operates and maintains the MS OS/2 *MULTIUSER* system by doing the following:

- System login, logout, and shutdown.
- Disk operation and maintenance.
- Querying, changing, and resetting status.
- System logging and auditing.
- Spooler operation.
- Problem determination.

## Appendices

- A: README Files.
- B: Different Ways to Set up Users and System Startup.
- C: Installing Async Ports for Citrix *MULTIUSER*.
- D: Setting up Specific Terminal Types.
- E: Cable Types and Configuration Implications.
- F: Setting up Specific Modem Types.

The System Administrator must be familiar with all the material in both the *Citrix MULTIUSER User's Guide* and the *Citrix MULTIUSER Command Reference*.

Use this guide to install MS OS/2 *MULTIUSER* on your system.

## Notational Conventions

To help you locate and interpret information easily, this guide uses specific typographic conventions and a standard syntax format and terminology. The following typographic conventions are used in this guide:

<u>Text Element</u>	<u>Notational Convention</u>
<b>KEYS</b>	Keys appear in boldface and uppercase.
<i>variables</i>	Variables are in lowercase italics.
USER INPUT	User input appears in uppercase and in a different typeface.

Text ElementNotational Convention

*FILENAMES,*  
*PROGRAM*  
*NAMES* and  
*DEVICE DRIVERS*

Names of files, programs (including applications), and device drivers are in uppercase italics.

COMMANDS,  
DIRECTORY  
NAMES, DRIVE  
NAMES, and  
UTILITIES

These are always in uppercase.

## Key Combinations

Key combinations and key sequences appear in the following format:

NotationMeaning

**KEY+KEY**

A plus sign (+) between keynames means you must press the keys at the same time. For example, "Press **ALT+ESC**" means that you press the **ALT** key and hold it down while you press the **ESC** key.

**KEY,KEY**

A comma (,) between keynames means you must press the keys in sequence. For example, "Press **ALT,SPACEBAR**" means that you press the **ALT** key and release it, and then press the **SPACEBAR** and release it.

<u>Notation</u>	<u>Meaning</u>
<b>DIRECTION</b> keys	Arrow keys on your computer keypad indicate DIRECTION. The name refers to the direction in which the arrow on the key points: <b>UP</b> , <b>DOWN</b> , <b>RIGHT</b> , or <b>LEFT</b> .

## Syntax Conventions

Syntax represents the order in which you must type a command-line command or utility name and any arguments and options that follow it. You may type commands, arguments, and options in either uppercase or lowercase letters. The following elements are used in syntax lines in this guide:

<u>Convention</u>	<u>Use</u>
<b>KEYWORD</b>	Elements shown in uppercase letters indicate the exact text to be entered. You may make your entry, however, either in uppercase or lowercase letters.
<i>variables</i>	Elements in lowercase italics are variables for which you must supply the text. For example, when <i>filename</i> appears, you should type the name of your file.
[ ]	Items in brackets are optional. To include the optional information described within the brackets, type only the information. Do not type the brackets themselves.

<u>Convention</u>	<u>Use</u>
	A vertical bar means that you must choose from the option on either side of it. For example, ON OFF means that you are to enter either ON or OFF.
<i>drive:</i>	Specifies a disk drive. You need to specify a drive name along with a <i>filename</i> only if you are using a file that is <b>not</b> on the current drive. The colon (:) must be typed as shown.
PATH	Specifies a complete directory path, using the following syntax:  [\\DIRECTORY...][\\DIRECTORY...]\\DIRECTORY  You need to specify a path along with a <i>filename</i> only if the file is not in the current directory.
...	An ellipsis indicates that an argument can be repeated as many times as necessary in a command line. Type only the information, not the ellipsis (...) itself.

# PART 1

## SYSTEM OVERVIEW

# CHAPTER 1

## INTRODUCTION



## OVERVIEW

This chapter provides you, the System Administrator, with an introduction to many of the MS OS/2 *MULTIUSER* concepts and functions. These are the concepts you will need to understand in order to install, configure, operate, and maintain the MS OS/2 *MULTIUSER* system.

Additional reading material for the System Administrator is provided on the installation diskettes. See the discussion on Readme files in Appendix A in order to find out what you need to do to get this additional information.

## USERS AND GROUPS

After installation, the most important job that the System Administrator has is to enable different people to use the system. In a standard multiuser configuration, every user logs into the system. In the process of logging in, the user gives the MS OS/2 *MULTIUSER* system the following information:

- User identification
- Password

The user identification is used by MS OS/2 *MULTIUSER* to determine which specific user execution and security environment is given to each person. This includes the initial default directory and the initial program. The *password* is used by MS OS/2 *MULTIUSER* as part of an overall security strategy. The *password* ensures that only the person authorized to use MS OS/2 *MULTIUSER* with that user identity can actually complete the user login process using that user identity.



The System Administrator is responsible for configuring new users into the system and setting up their initial execution environment. This initial execution environment is usually the Program Selector running in an initial directory, which is also the user's home directory.

The name that a user uses when logging in is called a *loginname*. When the System Administrator adds a new user to the system configuration, a new *loginname* is created for that user.

A *loginname* consists of two other names concatenated together. Those two names are *username* and *groupname*. A *loginname* is therefore also known as a *username.groupname*. When the System Administrator gives a person a *username*, that name can be the person's nickname followed by the first letter of the person's last name. For example, a good *username* for John Doe would be johnd.

The System Administrator must associate every *username* with a *groupname* in order to create a *loginname*. The System Administrator can also associate a *username* with a default *groupname*. A user can login with just a *username* if the desired *loginname* includes the default *groupname* for that *username*. If a user only has one *loginname*, that scenario is the most likely one.

The System Administrator can create a new group with a number of characteristics such as security class and resource management limits. A number of users can then be created in that group. Unless overridden, the group characteristics will become the user characteristics. Groups are used to create groupings of users with similar characteristics.

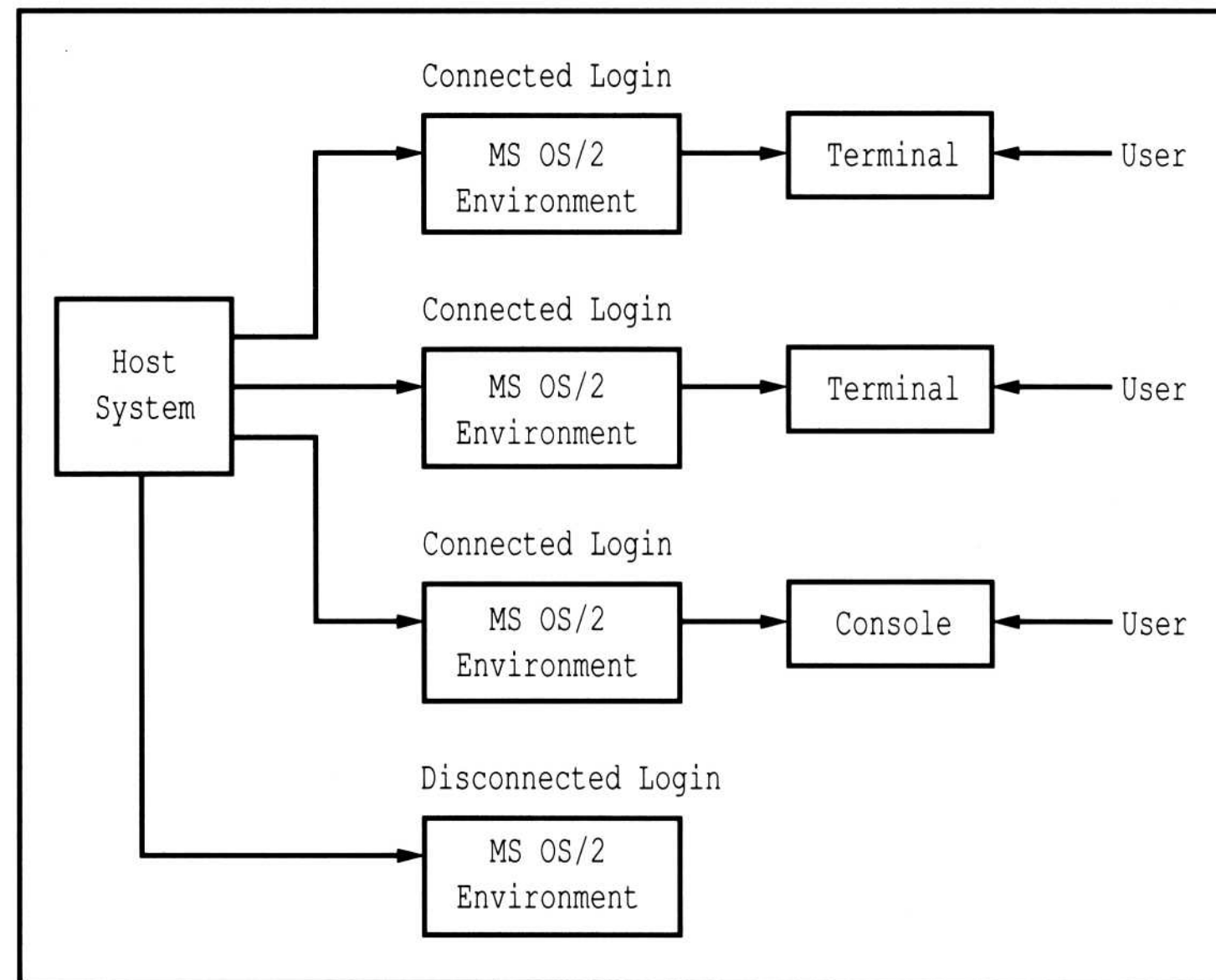
The same *username* can be associated with two different *groupnames* to create two different *loginnames*. This is useful if the same person needs to login under two different personalities. For example, one personality to run payroll and another personality to do order entry tasks.

The MS OS/2 *MULTIUSER* system installation procedure allows many users to be configured during the installation process. In addition, there is a preconfigured System Administrator loginname that the System Administrator can use. See Chapter 2, "System Installation," for additional information on System Installation.

After the system installation procedure has been completed, more users and groups can be added through the CONFIG utility. See Chapter 5, "Configuring Profiles," for additional information.

## LOGINS AND SESSIONS

After a user logs into MS OS/2 *MULTIUSER*, the user has an MS OS/2 text environment in which to perform tasks and run programs. This MS OS/2 environment is called a login. The login belongs to the *loginname*; that is, the *username* and *groupname* that the user provided when the user logged in. Refer to the following figure.



This means that security checks and resource management performed by the system for a login use the permission and resource characteristics assigned to the user's *loginname*. The login is connected (or attached) to the terminal at which the user logged in. Programs run by the user interact with the user through the display and keyboard of the terminal where the login is connected. The system considers these programs to be running in the context of the login.

Each time a user logs in, the system creates a new login for the supplied *loginname*. A *loginname* may have several logins active at the same time. For example, this situation occurs when one person logs in with the same *loginname* at more than one terminal without logging out.

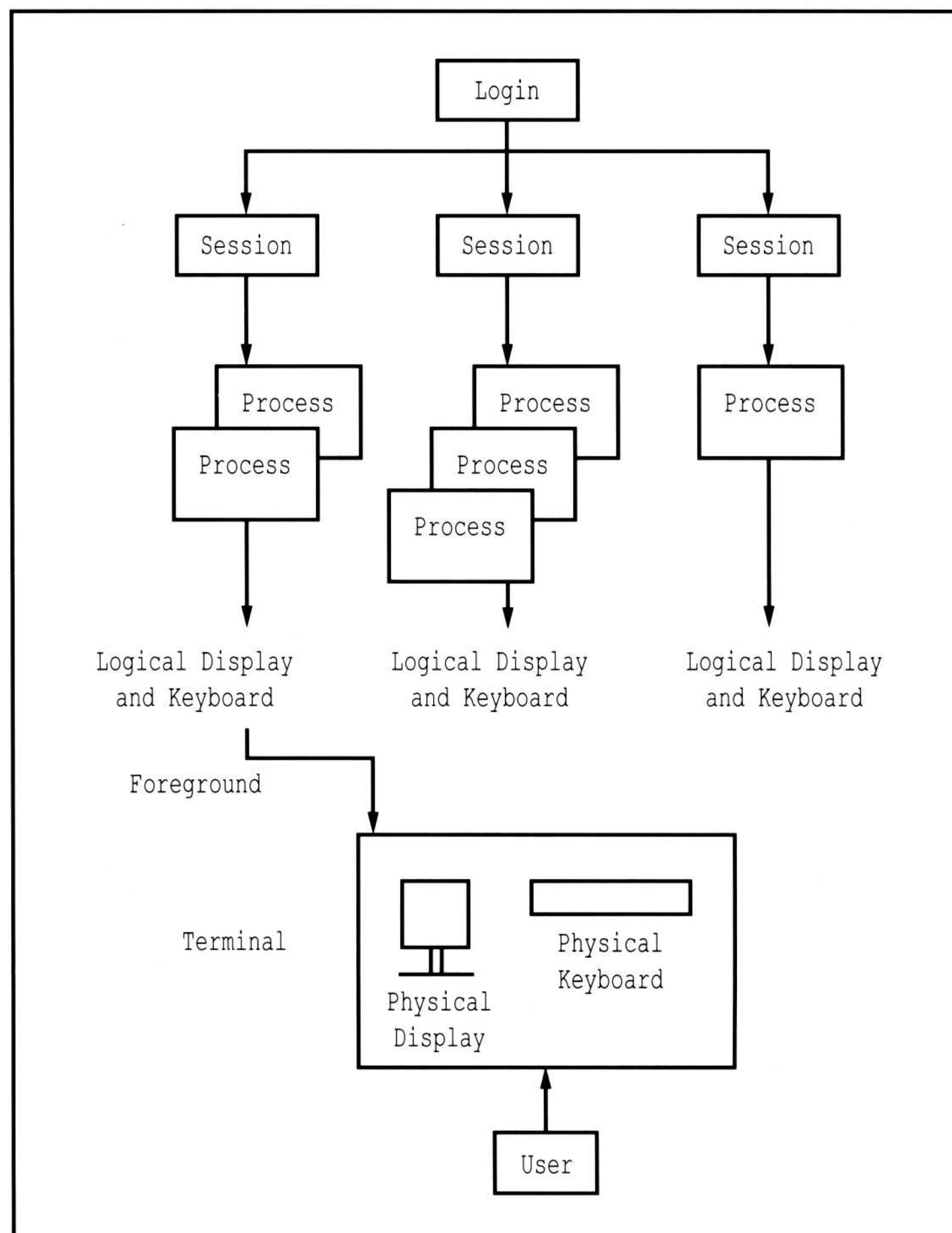
A disconnected login is a login that is not attached to a terminal. Programs in a disconnected login run in the background. A user may employ the DISCONN utility to disconnect the login from the terminal if the user needs to leave the terminal but keep the programs running. MS OS/2 *MULTIUSER* will automatically disconnect a login from its terminal when it detects that the terminal is no longer communicating with the system.

MS OS/2 *MULTIUSER* licenses the maximum number of simultaneous connected logins. See "Managing the Licensed Number of Users" in Chapter 12 for additional information.

System functions are performed within a login under a special *loginname*, SYSTEM.SYSTEM, reserved for MS OS/2 *MULTIUSER* control programs. This login is a disconnected login.

A login contains at least one session. Refer to the following figure. Every application program executes in the context of a session. A session represents a set of logical devices that a program needs to interact with the user, like a display and a keyboard. When a user logs in, the first session of the login is created for the initial program started for that *loginname*. This session is the foreground session of its login. In the foreground session, the logical devices of the session are associated with the physical devices of the user's terminal. Therefore, the user and the initial program interact through the terminal's display and keyboard. The user can start another session by using the **ALT-TAB** hotkey, the Session Create hotkey. This results in another session being created and a program (*CMD.EXE*, the command interpreter) being started. This session moves to the foreground and the previous session moves to the background.





The program in the first session continues to execute in the background. Programs execute regardless of whether they are in a background or foreground session. However, a program in a background session cannot interact with the user. Instead, a program in a background session can perform I/O to the session's logical devices but it will not receive actual input from the user until the user selects the session, which causes the session to become foreground (attached to the physical devices).



The user selects a session by using the **ALT+ESC** hotkey, the Session Switch hotkey. The **ALT+ESC** hotkey selects the next session in the set of sessions within the login and brings this session to the foreground. Session switching can also be performed using the Program Selector, which provides an interface to start programs in separate sessions.

Certain programs can be run completely in the background; that is, they perform functions without interacting with the user. A program of this type can be started in a session that is detached. A detached session is a background session that cannot be switched to the foreground.

A program running in a background session, regardless of whether the session is detached or not, can temporarily become the foreground session without user intervention. The background program temporarily creates a popup session to interact with the user.

While the popup is active, the program can perform I/O to the terminal devices. However, once the program ends the popup environment, the program returns to background execution. The user must satisfy the request of the program that has temporarily become the foreground session in order to resume using the previous foreground session.

The number of sessions that a login can hold and the number of simultaneous logins that a *loginname* can have will depend on the resource limits associated with the *loginname* that owns the login.

## SECURITY

A very important aspect of the administration of a multiuser system is to properly utilize its security features. The system security should keep users from interfering with each other and from damaging the system operation.

In the Users and Groups section of this chapter there is a discussion of the use of passwords. Passwords are available to ensure that only people who have permission can use (login to) the system. Password security is always enabled.

Another aspect of MS OS/2 *MULTIUSER* security is the ability to define Access Control Lists (ACLs) for certain system resources such as:

- Devices (for example, drive letters or character devices)
- Directories
- Files

As discussed in the HPFS section of this chapter, the High-Performance File System must be used on any fixed disk partitions for which file and directory security is desired.

An ACL can be thought of as a lock on its associated system resource. The ACL includes the information on who has the ability to open the lock and access the resource. MS OS/2 *MULTIUSER* security administration allows a *username*, *groupname*, or *loginname* to be given access or denied access to an ACL protected resource.

The System Administrator can decide to protect any device, disk directory, or file from one or a set of the users who can login to the system. Another way to view it is that the System Administrator can decide which users get access to which resources. For example, the System Administrator can create a directory on the disk and then only give himself or herself access to it.

The System Administrator can make securing system resources easier by setting up groups of users that have common security access privileges. Because ACL access can be associated with a *username*, *groupname*, or *loginname*, the System Administrator can create groups of users (who all have the same *groupname*) and then give them access or deny them access as a group.

Chapter 11 discusses what it means to install applications in a secure environment. The best configuration is one where users can use the application but cannot change the application or its associated files. Sometimes it may be necessary to give a program access to a resource that the user can not access. This can be done by registering the program with the REGISTER utility, thereby assigning it a unique security identity.

The security configuration facilities are always enabled in the system, as are login and *password* functions. However, run time access checking can be disabled. Disabling run time access checking means that regardless of what ACLs are present, no user will be denied access to any resource or function. Access checking can be disabled by entering Maintenance mode or by turning it off with the `SECURITY=OFF CONFIG.SYS` configuration command.

Every *loginname* in the system belongs to a security class. The class concept is independent from the group concept. There are four possible classes:

- Administrator
- Operator
- User
- Guest

Guest is the lowest security clearance and Administrator is the most privileged security clearance. In other words, if the system allows a User class to have access to a certain function, an Operator or Administrator class user will also have access. Users with the same *groupname* can be placed in different security classes by the System Administrator.

Certain user-oriented functions in the system (for example, many multiuser utilities) are secured. A user must belong to a certain level of security class in order to use the function. The SHUTDOWN utility is a good example of this. Someone in the User class should be denied access to the ability to turn the system off because this affects all of the users of the system. Many administration and operation functions are denied to *loginnames* in the User class. Security class access checking can also be turned off with the SECURITY=OFF configuration command.

Even if a user has access to a user-oriented function because the user is in the appropriate class, the user could be denied access to the system resources necessary to perform that function because of ACL processing.

See Chapter 7, "Configuring Security" for more information on MS OS/2 *MULTIUSER* security.



## EVENT LOGGING

MS OS/2 *MULTIUSER* provides a basic system event manager for security auditing and problem determination. This facility provides the ability to:

- Select events to log
- Maintain log files
- Selectively query any event log file

Events that can be selectively logged are:

- Security audit events
- System errors
- Terminal subsystem events
- Multiuser subsystem events
- Resource events
- Profile update events

A log file is created each time the system is initialized; log records that describe the selected events are placed in this file. The EVENTS utility can be used to query log data and close online log files. QUERY EVENTS will list which events are currently being logged and CHANGE EVENTS can temporarily change which events are being logged. The CONFIG SYSTEM utility is used to establish the standard set of events that are to be logged on a regular basis. For more information see Chapter 14, "System Event Logging."



System installation automatically enables system logging by placing a LOG=ON command in the *CONFIG.SYS* file. This is the recommended mode of operation; however, all event logging, including security auditing, can be completely removed by placing LOG=OFF in the *CONFIG.SYS* file. If this is done, no security audit data will be written to log files even though auditing was turned on by other means as described in Chapter 7 in the "Security Audit" section.

## RESOURCE MANAGEMENT

The System Administrator can use the resource management capabilities of MS OS/2 *MULTIUSER* to ensure a preset distribution of certain system resources across all the *loginnames* that are configured in the system. This capability gives the System Administrator the power to give every user in a system equal rights to the resources that the system can provide. The System Administrator can also choose to configure the system so certain users have a bigger claim to the available resources. A user running a word processor, a spreadsheet, and a personal program will probably need more system resources than order entry clerks who will run one predetermined program.

Resource Management is also responsible for creating per user login resource environments that allow applications to run as though there is only one user using them. See the description of System Global/User Login Global Shared Resources in this section.

All resources are managed only among those users currently logged into the system. Virtual memory, for example, is only of concern when users are logged onto the system. As memory is allocated on behalf of a user, checks are made to determine if the user is beyond his or her maximum limit for this particular resource.

The limits that are checked by the Resource Manager are initially set to defaults provided with the system. The System Administrator can query or set the defaults using utilities provided with the system. The System Administrator uses the QUERY, CHANGE, RESET, and CONFIG utilities to manage the resource allocation to the different *loginnames* in the system.

Resource limits can be disabled via the *CONFIG.SYS* command RESOURCE=OFF. The default when the system is installed is RESOURCE=ON. During system installation the System Administrator can disable resource management processing. Initial system limits are provided with the system. These system-wide defaults can be changed via the CONFIG SYSTEM utility.

Resource management processing is done on a per *loginname* basis. If the same user logs in under the same *loginname* more than once, the union of all the logins with the same *loginname* is used in whatever resource management processing decisions that are made in the system. If the same user logs in under two different *loginnames*, resource management considers those two logins as completely independent. Therefore, a user who logs in under two different *groupnames* can be given independent resource management parameters.

In order to simplify the following descriptions of minimum resource limits and maximum resource limits, the assumption is that a user has only one *loginname* and logs in once.

All resources have maximum allowed limits but some resources also have minimum guaranteed limits. These limits are described in the following sections.

## Maximum Resource Limit

As the name implies, the Maximum Resource Limit is the maximum amount of resources that a user can allocate. An attempt to allocate a resource (for example, virtual memory) past this limit will result in a popup message on your screen. Only a user in the Administrator or Operator privilege class can increase this value. If the limit is increased before the user replies to the popup, the user can retry the operation; otherwise the user's only recourse is to return the error to the application.

## Minimum Resource Limit

The Minimum Resource Limit is useful to guarantee resource availability for each user once he or she is logged into the system. It essentially reserves the amount of resources for the user, allowing no other user to allocate resources that would infringe on the user's minimum limit. As an example, user Joe could have a minimum limit of 1 megabyte (MB) of virtual memory. If Joe only uses 512K of his 1 MB minimum limit, there will be at least 512K of free memory in the system while Joe is logged in. Other users will get out-of-memory errors on memory allocations before Joe's 512K of virtual memory is used.

Minimum resource limits are only in effect when the user is logged on. For example, if a user's minimum memory limit is set to 1 MB and the physical or virtual memory is not available, the user will get an error message and the login will fail.

Not all resources have minimum limits.



## Resources Managed

The following resources are managed by Resource Management:

Amount of virtual memory	(Min / Max)
Number of threads	(Min / Max)
Number of system semaphores	(Min / Max)
Number of file handles	(Min / Max)
Number of logins for that loginname	(Max only)
Number of sessions per loginname	(Max only)

## System Global/User Login Global Shared Resources

In a multiuser system environment, software barriers must be put up between users, preventing one user login from adversely affecting the other user logins in the system. The resource management component of MS OS/2 *MULTIUSER* is responsible for creating a per user login resource environment that allows applications to run as though there is only one user using one copy of any given application. The System Administrator does not take any explicit action to enable this feature of resource management. The `RESOURCE=` statement in *CONFIG.SYS* does not affect this function of resource management, which is always active. As discussed below, certain classes of applications may require this function of resource management to be bypassed. This is also discussed further in Chapter 11, "Configuring Applications."

Resource management separates named shared resources (that MS OS/2 application programs use) into user login specific contexts. This prevents conflicts for resources when multiple users attempt to run the same application. As an example, if a word processor and spreadsheet application use named shared memory for a cut-and-paste buffer, other

users (running the same application) must have a different cut-and-paste buffer for their work. If they don't, one user would be able to cut-and-paste another user's data.

This separation of named shared resources actually applies to each user login. If you login to the system more than once using the same *loginname*, your cut-and-paste buffers will be unique to each login. This is different from minimum and maximum guaranteed limits, which apply to the union of all the logins under the same *loginname*.

Some MS OS/2 applications have been designed with multiuser considerations in mind. Client/Server applications, whose execution can be distributed over a LAN or run completely in a single system, are examples of a class of existing MS OS/2 applications that have multiuser considerations. For these applications, it may be important for the names of named shared resources to be the same for the client and server portions of the program. In other words, a system global context is needed instead of a user login context. For these programs the REGISTER utility can be used to mark the appropriate *.EXE* or *.DLL* file(s) as System Global. See Chapter 11, "Configuring Applications" for further information.

## PRINTING AND SPOOLING

Printers are shared among many users and processes through printer queues. A printer queue stores print jobs (data destined for the printer) as they are submitted to the printer, and routes them to the printer when the printer becomes available. A printer queue can route jobs to one or more printers.



Printer queues are spooled. That is, each print job is stored in a queue on disk until a printer is ready. A program that sends a print job to a printer can continue working without waiting for the printer to become free. If many programs are printing at the same time and the printer is not spooled, the printer will print unpredictable output.

Print jobs can be arranged in the queue in two ways. They can be ordered by FIFO (first in-first out) or print job size. If ordered by FIFO, the print jobs are printed in the order in which the spooler received them. If ordered by size, smaller jobs are printed before larger ones.

The spooler is the process that intercepts data heading for the printer, spools the data, and manages the queues. All spooling activity is done in the background, transparent to the user.

The spooler also keeps a "shadow" of each print job in the queue(s) until the print job has been totally printed. If for some reason the system shuts down when there are still print jobs in a queue, the next time the spooler is run it will be able to continue processing these print jobs from where it left off.

A printer queue can cause a separator page to print before each job. The separator page usually tells what user submitted the print job, when it printed, and so forth. See Chapter 8, "Configuring Printing and Spooling" for more information.

To configure and manage queues and print jobs, use the SPOOL utility. See Chapter 8, "Configuring Printing and Spooling" and Chapter 15, "Spooler Operation" for more information.

## TERMINALS

Users interact with the MS OS/2 *MULTIUSER* system via terminals. A terminal consists of a keyboard, a display, and a communications connection to the host computer. MS OS/2 *MULTIUSER* supports many different types of terminals.

There are several steps to follow to connect a terminal to the system. The terminal itself must be setup to have the correct characteristics. MS OS/2 *MULTIUSER* must also be configured properly for the terminal. This includes identifying what communications port the terminal is connected to. Some configuration parameters may depend on what cable and terminal you will be connecting.

You may also need to install terminal subsystems. Terminal subsystems consist of the hardware adapter that the terminal connects to as well as the terminal subsystem software necessary to have the hardware adapter work with MS OS/2 *MULTIUSER*.

Dialing into the computer over a telephone line from remote terminals is supported. You will need to configure a given communications connection appropriately for the modem and communications characteristics of the given dial-in port.

As the System Administrator, you can choose to configure a terminal so that when the system detects that it is powered on and connected, a login prompt will appear. Another option you have is to configure the terminal to power on with a specific program running in a specific user login context. This feature is very useful for configuring turnkey operations with terminals running a closed set of functions.

Chapter 6, "Configuring Terminal Subsystems and Terminals;" Appendix C, "Installing Async Ports for Citrix *MULTIUSER*;" Appendix D, "Setting Up Specific Terminal

Types;" Appendix E, "Cable Types and Configuration Implications;" and Appendix F, "Setting Up Specific Modem Types" have a complete description on how to install and configure terminals and terminal subsystems.

The system console is a terminal that is physically part of the host computer. You must power on the host computer for MS OS/2 *MULTIUSER* to function. Normally, the system console will function like any other terminal. One difference is that you can configure the system console reboot hotkeys to reboot the entire system instead of just the user login with the *CONFIG.SYS* parameter REBOOT.

## THE HIGH-PERFORMANCE FILE SYSTEM (HPFS)

A feature of MS OS/2 *MULTIUSER* is its capability for multiple file systems to co-exist simultaneously and support the different block mode devices (such as fixed disks) that the system supports. For example, the original file allocation table (FAT) file system is always present in the system and supports diskettes as well as any FAT partitions that may exist on the fixed disks (but are not required). The High-Performance File System (HPFS) is an installable file system that is always shipped with and installed on MS OS/2 *MULTIUSER*. HPFS supports the required HPFS fixed disk partitions(s). You can install additional file systems as well. In MS OS/2 *MULTIUSER*, an additional file system is referred to as an installable file system (IFS).

The HPFS installable file system is always installed during MS OS/2 *MULTIUSER* installation and is required for use on the primary partition of the boot drive. The simplest approach is to have only one partition per fixed disk and have HPFS support all the fixed disks. This will ensure that security can be enabled for all your fixed disks.



HPFS completes the tasks of writing to and reading from the fixed disks much faster than the FAT file system can. When naming files and directories, HPFS allows the use of up to 254 characters. (Some of those characters are not allowed in the FAT file system.) Longer names and using a wider variety of characters are useful for creating descriptive file names. HPFS also offers expanded naming conventions and wild card matching capabilities. See Chapter 12 for additional information.

HPFS supports lazy writing techniques to speed up programs that read from and write to the disk. In most cases, when a program needs to write information to the fixed disk, it is not necessary that the information be written immediately. HPFS will temporarily store information that a program is writing to disk in memory. Later, when the disk would otherwise be idle, HPFS writes the information to the disk as a background activity. This is called lazy writing. The program can execute much faster because as soon as the data is in memory, the write operation is completed from the perspective of the program.

MS OS/2 *MULTIUSER* requires the use of HPFS when file and directory security is desired on the files and directories in a fixed disk partition. In other words, an Access Control List (ACL) can be placed on an HPFS file but not on a FAT file. See Chapter 7, "Configuring Security" for details and recommendations.

## SYSTEM USER/TERMINAL STARTUP OPTIONS

The Administrator has a tremendous amount of flexibility in determining how the MS OS/2 *MULTIUSER* system is configured from the perspective of the user structure and startup. The way the system should be configured is determined by

the desired end result. The following is an overview of some of the existing options. The mechanics on how to achieve the capabilities discussed in this section are covered throughout this book.

A terminal can be configured to initialize with a login prompt or to automatically login a *loginname*. The startup options selected for a *loginname* determine what happens next.

When a *loginname* logs into the system, the initial default directory can either be the *username*'s home directory or a preconfigured initial working directory. Multiple *loginnames* can use the same initial working directory as the default directory if that is the desired effect.

The *loginname*'s initial program can either be the Program Selector, the command interpreter, or a specific program. By choosing the initial program appropriately (or by using the appropriate security configuration), the System Administrator can force a specific user or terminal to execute only a fixed program or set of programs.

The different startup files can be used to customize to whatever final result is desired.

- *CONFIG.USR*, which is processed by the system when a user logs into the system. It is found in the username's home directory.
- *STARTUP.CMD*, which is processed by the Program Selector when the Program Selector starts.
- Any batch file that is associated with the starting of CMD, the command interpreter.



A typical multiuser open ended configuration would not use very much of the flexibility discussed above. Every terminal would be configured to come up with a login prompt. The initial program for each user would be the Program Selector and the initial default directory would be the home directory. The Program Selector might have some programs pre-setup for each user. If standard security is installed, users could not interfere with each other's disk directories.

Many other configuration possibilities exist that are different from the typical configuration discussed above. See Appendix B, "Different Ways to Set Up Users and System Startup" for a description of some other possible configuration end results that may be desired and a description of how the options discussed in this section can be used to achieve those desired end results.



# PART 2

## SYSTEM INSTALLATION

# CHAPTER 2

## SYSTEM INSTALLATION

## INTRODUCTION

This section describes some options that you should be familiar with before you install MS OS/2 *MULTIUSER*. Following these options are step-by-step instructions for setting up and starting MS OS/2 *MULTIUSER* on your computer.

## BEFORE YOU INSTALL MS OS/2 *MULTIUSER*

While you are setting up MS OS/2 *MULTIUSER*, you will be asked to choose particular options or answer questions about the kind of hardware you have. It may speed up your installation if you take a moment now to familiarize yourself with the following items:

- You must use the High-Performance File System (HPFS) formatted by Citrix *MULTIUSER* on your primary partition of your first fixed disk. During installation, if your primary partition is not HPFS or if it was not previously formatted by Citrix *MULTIUSER* installation, you are required to reformat your fixed disk for Install to continue. Backup any programs or data you want to save from your primary partition if the installation procedure will cause a reformat.
- The amount of space available on your fixed disk. Normally, only HPFS is installed on the fixed disk. If you must install both HPFS and FAT or if your fixed disk is already partitioned, see the next section.
- If you are going to attach devices or add terminals to the planar serial port, add this support when you are prompted.
- You will be prompted for the type of display adapter.

- You can add up to ten users plus one Administrator during installation.
- You can choose to disable security access checks. However, before you do this, familiarize yourself with the security facilities.

## INSTALLING FILE SYSTEMS

If your first fixed disk is not partitioned, the simplest installation procedure you can follow is to have the installation program set up a primary partition for your first fixed disk that takes up all the space on the disk. The result is a single partition on your first fixed disk. If you follow this procedure, your fixed disk is ready to use after installation is complete. Therefore, if you want a single primary partition, go directly to the section "Running the Installation Program."

If your first fixed disk is already partitioned, read the next section, "Installing with Multiple Partitions," to see if you can keep the existing partition structure. If you can keep the existing partition structure and you want to keep the existing partition structure, go directly to the section "Running the Installation Program." Otherwise, follow the appropriate procedure in the next section, "Installing with Multiple Partitions" to install with a single partition or multiple partitions on your first fixed disk.

If you want to have multiple partitions on your first fixed disk, go to the next section, "Installing with Multiple Partitions."

No matter how you setup your first fixed disk, if you have more than one fixed disk, read the section, "Installing with Multiple Fixed Disks."



## Installing with Multiple Partitions

If you want to install your system with multiple partitions (drive letters) on your first fixed disk, or if your first fixed disk has existing partitions, read this section before you install MS OS/2 *MULTIUSER*.

If your computer's first fixed disk is already divided into multiple partitions and you wish to leave it unchanged, you may not need these instructions. However, in order to keep your partitions as they are, the primary partition (drive C) must be large enough to satisfy the MS OS/2 *MULTIUSER* system requirements. It is recommended that the primary partition be at least 20 megabytes to accommodate system configuration files and dynamic system files like the swap and log files. In addition, drive C must be formatted for HPFS. If the partition is not large enough, continue to read these instructions. If it is large enough but not formatted for HPFS or it has never been formatted by Citrix *MULTIUSER* installation, you should back up your data that is on drive C. If you want to keep the existing partition structure, go directly to the section "Running the Installation Program."

The following instructions are for two different fixed disk configurations. Select the description that best fits your current status and desired results.

If your computer's first fixed disk is not partitioned and you wish to have multiple partitions on your first fixed disk, do the following:

1. Complete the instructions in the section, "Running the Installation Program." During installation, set up a primary partition that is smaller than the total disk space but large enough to satisfy the MS OS/2 *MULTIUSER* system requirements. The primary partition is formatted as HPFS.

2. After installation, run the FDISK utility, set up an extended partition and logical drive, and use the FORMAT utility to format for the file system you wish, HPFS or FAT.

If your computer's first fixed disk is partitioned and you don't want to keep the partition structure, follow these steps:

1. Back up any programs or data you want to save onto floppy disks.
2. Use the FDISK /D utility to delete the partition(s). (Insert the MS OS/2 *MULTIUSER* Install disk, restart your computer, insert disk 1 when prompted, and press **ENTER**. When the first panel appears, press **ESC** and type FDISK /D.)
3. If you want a single primary partition on your first fixed disk, go directly to the section, "Running the Installation Program." If you want multiple partitions, continue to step 4.
4. Follow the instructions in "Running the Installation Program." During installation, set up a primary partition that is smaller than the total disk space. It is formatted as HPFS.
5. After installation, run the FDISK utility, set up an extended partition and logical drive, and use the FORMAT utility to format it for the file system you desire, HPFS or FAT.

See Chapter 3 on "Setting Up Your Fixed Disk Partitions" and the *Citrix MULTIUSER Command Reference* for the details on the FDISK and FORMAT utilities.

## Installing With Multiple Fixed Disks

The installation procedure can be used to partition and format your entire first fixed disk as a single primary partition. The installation program does not operate on any other fixed disks that your hardware configuration has. Therefore, if you have more than one fixed disk (and they are not already partitioned and formatted), you must use FDISK and FORMAT to partition and format all your other fixed disks after you have completed the installation procedure and started MS OS/2 *MULTIUSER*.

## RUNNING THE INSTALLATION PROGRAM

An installation disk, included with your MS OS/2 *MULTIUSER* package, makes it easy for you to install MS OS/2 *MULTIUSER* on your computer by following these steps:

1. Turn on your computer.
2. Put the disk labeled Install in drive A (drive A must be a high-density drive).
3. Press and hold down the **CTRL+ALT+DEL** keys.
4. Release all three keys and then wait for the first screen of information to appear. Your computer may take a moment to read the Install disk.
5. Follow the instructions you see on each screen.

After you have completed a screen, press the **ENTER** key to move to the next screen. To return to the previous screen, press the **ESC** key. If you make a mistake, press the **ESC** key repeatedly until you return to the screen where you made the mistake. When you have corrected your mistake, press the **ENTER** key repeatedly until you return to the screen where you want to continue working.



## Getting Help

If you need help while installing MS OS/2 *MULTIUSER*, press the **F1** function key. Pressing **F1** displays Help information for your current screen.

## AFTER MS OS/2 *MULTIUSER* INSTALLATION IS COMPLETE

During installation, *username* CITRIX is automatically created as a member of the group that also has the *groupname* CITRIX. User CITRIX.CITRIX is an Administrator who is configured to expire five days after installation. The account expiration date of user CITRIX.CITRIX can be changed using the configuration utilities described in Chapter 5. If you did not modify the security settings during installation, there will also be a *username* ADMIN, which is a member of group WRKGRP and is also an Administrator. The default *password* for both CITRIX.CITRIX and ADMIN.WRKGRP is "citrix" (lowercase) unless you changed the *password* for ADMIN.WRKGRP during installation. If you change the *password*, it is changed for both *loginnames*.

During installation, certain system configuration settings are set in the *CONFIG.SYS* file. For more information, see Chapter 9, "Using System Configuration Commands."

## Starting MS OS/2 *MULTIUSER*

To start MS OS/2 *MULTIUSER*, simply turn on your computer, or if your computer is already turned on, press **CTRL+ALT+DEL** to restart it. Each time you turn on your computer or restart it, MS OS/2 *MULTIUSER* starts automatically.

The following chapters describe how to add terminals, users, and security to your system. Refer to these sections to make changes to the installation defaults.





# CHAPTER 3

## SETTING UP YOUR FIXED DISK PARTITIONS

## INTRODUCTION

Before you use a fixed disk, you need to perform certain setup tasks. These setup tasks may include any of the following:

- Displaying partition data or logical-drive information.
- Creating or deleting a primary partition.
- Changing the start-up partition.
- Creating or deleting extended partitions and logical drives.
- Formatting the fixed disk.

This section describes the tasks involved in setting up your fixed disk; the tool you use to perform these setup tasks is the Fixed Disk Utility.

## CONFIGURING A FIXED DISK

The FDISK utility sets up your fixed disk for use with MS OS/2 *MULTIUSER*. You must always run FDISK on a fixed disk before using it for the first time. The MS OS/2 *MULTIUSER* installation program does this for the primary partition of your first fixed disk, but later you might want to run FDISK to change the configuration of your first fixed disk. After installing MS OS/2 *MULTIUSER*, you must run FDISK and FORMAT for any (unpartitioned and unformatted) additional fixed disks you have or for creating an extended partition on your first fixed disk. FDISK will allow you to create, delete, and modify partitions and logical drives on up to seven disks.

**NOTE:** FDISK can be invoked from the command prompt. FDISK must be invoked in maintenance mode if partitions are to be modified. FDISK can also be invoked after booting from the Install diskette. To run FDISK from the Install diskette, insert the MS OS/2 *MULTIUSER* Install diskette, restart your computer, insert disk 1 when prompted, and press **ENTER**. When the first panel appears, press **ESC**. You may then type FDISK at the A:> prompt.

FDISK is invoked during system installation to create a primary partition if one does not exist. You will be queried as to the size of the partition. You may select a default size (which would consist of the entire disk) or you may specify a size in megabytes. FDISK may be invoked during installation to delete all primary partitions, extended partitions, and logical drives, using FDISK /D.

You should run FDISK if you need to change the size or number of partitions on your fixed disk or if you must restore lost partition information. In addition, you can run FDISK to find out the way your fixed disk is configured.

In the following sections, specific examples are used to illustrate how FDISK options work. Your computer may display different values when you run FDISK, depending upon the size of the fixed disk installed in your system.

**WARNING:** Reconfiguring your disk by using FDISK destroys all existing files. Be sure to have a backup copy of all the files on your disk before you run FDISK.

## Partitions and Logical Drives

A fixed disk can be organized into separate sections called partitions. You can set up one primary and one extended partition. The extended partition is optional. The primary partition is drive C; it is where the start-up files for MS OS/2 *MULTIUSER* must reside. It can also contain user files and

directories. An extended partition contains user files or directories, but it cannot contain MS OS/2 *MULTIUSER* start-up files. Within an extended partition, you can have one or more logical drives, each having its own drive letter (for example, D, E, and F).

You will probably want to set up the entire fixed disk as the primary partition. That way you can gain access to any file and directory on the disk without having to change to a different logical drive. There, however, may be some instances when you may need to set up and use an extended partition and logical drives.

## Setting Up Your Fixed Disk

If your fixed disk is being configured for the first time, you may need to perform the following steps. The MS OS/2 *MULTIUSER* installation program performs these steps for you. These steps are described in the following sections.

1. Setting up the primary (start-up) partition.
2. Optionally, setting up the extended partition and logical drives.
3. Restarting your system to make the changes permanent.
4. Formatting drive C and any logical drives you have created.

**NOTE:** Drive C must be formatted using the High-Performance File System (HPFS).

To find out whether your fixed disk has already been configured for MS OS/2 *MULTIUSER*, try to start MS OS/2 *MULTIUSER* from your fixed disk. If it starts, your fixed disk is both configured and formatted, and the MS OS/2 *MULTIUSER* system files are on the disk. If MS OS/2 *MULTIUSER* does not



start, your disk is not formatted to start MS OS/2 *MULTIUSER*, but it may have been configured. Check to see whether the disk has been configured by using the FDISK command and then selecting the Display the Partition Data option. This procedure is discussed in greater depth later in this chapter.

## Starting the FDISK Program

The FDISK utility uses menus that lead you through each procedure.

To start FDISK if MS OS/2 *MULTIUSER* is installed on your fixed disk, type FDISK at the prompt and press **ENTER**. Partitions can only be modified in maintenance mode.

If MS OS/2 *MULTIUSER* is not installed on your fixed disk, follow these steps:

1. Place the MS OS/2 *MULTIUSER* Installation disk in drive A.
2. Turn on your computer. If your computer is already turned on, restart your computer by pressing **CTRL+ALT+DEL**.
3. Insert disk 1 when prompted and press **ENTER**.
4. When the first panel appears, press the **ESC** key. This takes you to the command prompt.
5. Type FDISK at the prompt and press **ENTER**.

In response to your typed command, the FDISK utility displays the FDISK Options menu on your screen. This menu lists five options (if your computer has only one fixed disk, the fifth option will not appear on your screen).

## FDISK Options

Choose one of the following:

1. Create a partition or a logical drive
2. Change the active partition
3. Delete a partition or a logical drive
4. Display the partition data
5. Select Next Fixed Disk Drive

Enter choice: [1]

Press Enter to continue or  
Esc to return.

Most of the FDISK menus display a default value. To choose the default value, press **ENTER**. To choose another value, just type the value you want and press **ENTER**.

The following sections describe each of the FDISK menu options and show the menus and other information they display. To return to MS OS/2 *MULTIUSER* from the FDISK Options menu, press the **ESC** key. You can also use the **ESC** key to return to the FDISK Options menu from any of the other FDISK menus.

To quit FDISK, return to the FDISK Options menu and press **ESC**. If you have created any partitions or logical drives, or if you have changed the active partition, you'll see the following message displayed on your screen:

```
The fixed disk has been updated. The system should
now be restarted. Press CTRL+ALT+DEL.
```

You must restart your system to make the changes permanent. In addition, if you have created any partitions or logical drives, you must format them with the **FORMAT** utility before copying any files to them.

## Creating the Primary Partition

The first step in configuring a fixed disk is to create a primary partition. You will most likely want your entire fixed disk to be in the primary partition, although this is not required.

When you create a primary partition, no partitions must currently exist on the fixed disk. This means that if you want to change the size of the primary partition, you must first delete existing partitions and logical drives, then create a new primary partition (see "Deleting a Partition" later in this chapter for instructions on deleting partitions). You must create a primary partition before you can create an extended partition. This restriction applies only to the first (boot) fixed disk.

To create the primary partition, select option 1 from the FDISK Options menu and press **ENTER**. FDISK displays the following screen:

```
Create a partition

Choose one of the following:

1. Create primary partition
2. Create extended partition

Enter choice: [1]
Press Esc to return to FDISK Options
```

To create the primary partition, press **ENTER** to accept the default selection (option 1).

The Create Primary Partition menu appears next. You will see the following message:

```
Do you want to create an active partition of the
maximum size (Y/N) .....? [Y]
```

If you want to have the entire fixed disk in the primary partition, press **ENTER** to accept the default selection (Y).

If you want to create a primary partition that is smaller than the maximum size, type N for no. The FDISK utility displays the maximum capacity of your fixed disk (in cylinders) and prompts you for the size you want your primary partition to be.

Create Primary Partition

The maximum capacity of the fixed disk is 732 cylinders.

Enter partition size.....: [ 730]

No partitions are defined

Press Esc to return to the FDISK Options

Type the size of the partition (in cylinders) and press **ENTER**.

FDISK then displays status information about the partition it just created and tells you that the primary partition has been created by displaying the following:

Create Primary Partition

Partition	Status	Type	Start	End	Size
1 C:	A	PRI DOS	2	601	600

The maximum capacity of the fixed disk is 732 cylinders.

The primary partition is created.

Press Esc to return to the FDISK Options.

You'll see the number and drive, status, and type of the partition; the partition's starting and ending cylinder numbers; and the partition's size (in cylinders). For a detailed description of each of these fields, see "Displaying Partition Data" later in this chapter.



After you create the primary partition, press **ESC** to return to the FDISK Options menu. If you want to create an extended partition, see the instructions found in the following section. Otherwise, you should restart your computer to activate the new partition. Before MS OS/2 *MULTIUSER* can use your fixed disk partition, you need to format the disk. For example, to format the primary partition on your second fixed disk, type:

```
FORMAT D: /FS:HPFS
```

Remember that formatting your disk destroys all data on it. Make backup copies of your disk as necessary.

## Creating an Extended Partition

You may choose to create a primary partition that is smaller than the maximum size and then use the remainder of the disk as an extended partition.

To create an extended partition, return to the FDISK Options menu and select option 1.

The Create Partition menu will appear on your screen:

```
Create Partition
```

1. Create primary partition
2. Create extended partition

```
Enter choice: [1]
```

```
Press Esc to return to the FDISK Options
```

Select option 2 to create an extended partition.

When you select option 2, the FDISK utility displays status information about the primary partition and prompts you as follows:

### Create Extended Partition

Partition	Status	Type	Start	End	Size
1 C:	A	PRI DOS	2	601	600

The maximum capacity of the fixed disk is 732 cylinders.

Enter partition size..... [ 130]

Press Esc to return to the FDISK Options

The partition size shown in square brackets is the number of cylinders left for your extended partition. In most cases, you'll want to use the maximum value. Press **ENTER** if you want this value; otherwise, type the size (in cylinders) that you want for the partition, then press **ENTER**.

FDISK then displays status information about the primary partition and the extended partition that you just created. To display the Create Logical Drive(s) menu, press **ESC** once; to return to the FDISK Options menu, press **ESC** twice.

## Creating Logical Drives on the Extended Partitions

After you create an extended partition, you should create logical drives for it. You may designate the entire extended partition as one logical drive or you may divide it into many logical drives. If you have only physical drive C, the first logical drive you may designate is called drive D, the second is called drive E, and so on. Logical drives are a good organization tool. You could, for example, have different applications and their data files on separate logical drives.

If you are adding logical drives to an existing partition, select option 1 from the FDISK Options menu to display the Create Partition menu, then select option 3 to display the Create Logical Drive(s) menu. If you are creating a logical drive for a new extended partition, this menu is automatically displayed after you create the new extended partition.

Create Logical Drive(s)

The total partition size is 130 cylinders.

The maximum contiguous space available for the logical drive is 130 cylinders.

Enter the size of the logical drive.....: [ 130]

Press Esc to return to the FDISK Options.

If you are adding logical drives to the extended partition, FDISK displays the current drive assignments as well as prompting you for new logical drive information. For example, if logical drive D already exists, you might see the following:

Create Logical Drive(s)

Drive	Start	End	Size
D:	602	651	50

The total partition size is 130 cylinders.  
The maximum contiguous space available for the logical drive is 80 cylinders.

Enter the size of the logical drive.....: [ 80]

Press Esc to return to FDISK Options.

The value in square brackets is the number of unused cylinders in the extended partition. Press **ENTER** to accept the default value listed in the square brackets. Otherwise, type in the number of cylinders for the logical drive and press **ENTER**.

If any cylinders remain in the extended partition, you will be prompted again for the size of the next logical drive. The FDISK utility will continue to prompt you until you run out of space in the extended partition, use all available drive letters (D through Z), or until you press **ESC** to return to the FDISK Options menu.

After you create logical drives in the extended partition, you should restart your computer by pressing **CTRL+ALT+DEL**. This makes the logical drive information permanent. Then, so that MS OS/2 *MULTIUSER* can use your fixed disk, format your disk by typing the following:

```
FORMAT drive: [/FS:HPFS]
```

**NOTE:** When you format any logical drive, you have the option of formatting for FAT or HPFS file systems. For the primary partition of the first fixed disk, you must choose HPFS. (This is done automatically during System Install.) Any drive you wish to have security on must be formatted by HPFS.

For example, if you have partitions for both some other operating system and MS OS/2 *MULTIUSER* on your disk, the Change Primary Partition screen might look like this:

```
Change the active partition
```

Partition	Status	Type	Start	End	Size
1 C	A	PRI DOS	2	601	600
2	N	OTHER.OS	602	731	130

```
Total disk space is 732 cylinders.
```

```
Enter the number of the partition you
want to make active.....: [1]
```

```
Press ESC to return to FDISK Options.
```

Type the number of the partition that you want to make active and press **ENTER**. The default setting is the number of the currently active partition.

If your fixed disk contains only MS DOS and MS OS/2 *MULTIUSER* partitions, FDISK displays the following message instead of prompting you for the partition that you want to activate:



The only start-up partition on Drive 1 is already marked active.

Press ESC to return to FDISK Options.

**NOTE:** If you want to run an operating system other than MS OS/2 *MULTIUSER* from your fixed disk, see the other operating system's manual for instructions on how to install and configure that operating system.

## Deleting a Partition

The FDISK utility lets you delete a primary partition, extended partition, or logical drive. To delete a primary partition, you must first delete the extended partition. To delete an extended partition, you must first delete any logical drives on it.

The FDISK utility doesn't allow you to change the size of a partition. Therefore, if you need to change the size of a partition, you must first delete the partition, then create a new partition that has the new size.

**NOTE:** Be sure to make backup copies of all files before you delete the drive. When FDISK deletes a logical drive or partition, the data on it is destroyed.

If you select option 3 of the FDISK Options menu, the FDISK utility displays the Delete Partition menu:

Delete Partition

1. Delete Primary partition
2. Delete Extended partition
3. Delete the logical drive(s) in the Extended partition

Enter choice: [1]

Press ESC to return to FDISK Options.



Option 3 will not appear if you have not created logical drives.

Type the number of the option you want and press **ENTER**. If you've chosen to delete a primary or an extended partition, the next menu shows the status of that partition. If you are deleting a logical drive, the drives currently defined are displayed.

## Deleting a Logical Drive

To delete a logical drive, select option 3 from the Delete Partition menu. The FDISK utility then displays status information about the existing partitions and prompts you as follows:

Delete the Logical Drive

Drive	Start	End	Size
D:	602	651	50
E:	652	731	80

Total partition size is 130 cylinders.  
Warning! Data in the logical drive will be lost.  
What drive do you want to delete.....? [ ]

Press ESC to return to FDISK Options

Type the letter of the drive you want to delete and press **ENTER**. The FDISK utility displays the following message:

Are you sure.....? [N]

If this logical drive contains valuable data that you have not made backup copies of, press **ENTER**. This stops FDISK from deleting the logical drive.

To delete the drive, type **Y** for yes. After deleting a drive, FDISK will continue to prompt you for more logical drives to

delete, until all drives are deleted or until you press **ESC**. Once you press **ESC**, status information for the remaining logical drives is displayed.

If you do not delete logical drives in the reverse order in which they were created, the remaining drives are reordered starting from drive D. For example, if you had created drives D, E, and F and you delete drive D first, drives E and F will be reordered and renamed drive D and drive E.

## Deleting the Extended Partition

To delete the extended partition, select option 2 from the Delete Partition menu. Partition status information appears on the screen and you are prompted as follows:

Delete Extended Partition

Partition	Status	Type	Start	End	Size
1 C	A	PRI DOS	2	601	600
2	N	EXT DOS	602	731	130

Warning! Data in the Extended partition will be lost.

Do you want to continue.....? [N]

If you want to prevent the partition from being deleted, press **ENTER**. You will then be returned to the FDISK Options menu.

If you want to delete the partition, type **Y** for yes. A message will appear on the screen, confirming that the partition has been deleted. It will also provide partition status information. Notice that only the primary partition now exists. To return to the FDISK Options menu, press **ESC**.

## Deleting the Primary Partition

To delete the primary partition, select option 1 from the Delete Partition menu. Partition status information appears on the screen and you are prompted as follows:

Delete Primary Partition.

Partition	Status	Type	Start	End	Size
1	C: A	PRI	DOS	2	601 600

Warning! Data in the primary partition will be lost.

Are you sure you want to continue .....? [N]

Press ESC to return to FDISK Options

If you do not want to delete the primary partition, press **ENTER** to accept the default value (no).

**NOTE:** You may only delete the boot primary partition (logical drive C) by running FDISK from the installation diskette (see "Starting the FDISK Program").

## Displaying Partition Data

If you choose option 4 on the FDISK Options menu, FDISK displays a screen that contains information about each of the partitions on your fixed disk.

The Display Partition Information screen contains the following information:

Display Partition Information

Partition	Status	Type	Start	End	Size
1	C: A	PRI	DOS	2	601 600
2		N	EXT DOS	602	731 130

Total disk space is 732 cylinders.

```
The Extended partition contains logical drives.  
Do you want to display the logical drive  
information? .....? [Y]
```

Press ESC to return to FDISK Options.

This screen identifies the partitions on your disk. It shows the number of each partition, its status, and its type. The screen also shows each partition's starting and ending cylinder numbers, in addition to its size (in cylinders). The following list explains the fields of partition information you will see:

- The Partition field tells you the partition number and drive name. The primary partition is drive C if the first fixed disk is selected.
- The Status field tells you whether the partition is active (A) or not active (N). The primary partition of the first fixed disk must be active if you want to start MS OS/2 *MULTIUSER* from it.
- The Type field tells you whether this is the primary (PRI DOS) or extended (EXT DOS) partition. If you have installed another operating system, its name will appear in this field.
- The Start and End fields tell you the cylinder numbers where each partition starts and ends, respectively.
- The Size field tells you the total size of the partition (in cylinders).

If you have an extended partition, the FDISK utility asks whether you want to see information about that partition's logical drives. Type **Y** for yes and press **ENTER** to display a screen similar to the following:

Drive	Start	End	Size
D:	602	651	50
E:	652	731	80

Press ESC to return to FDISK Options

Press the **ESC** key to return to the FDISK Options menu.

## Selecting the Next Fixed Disk Drive

The fifth option, Select Next Fixed Disk Drive, appears on the FDISK Options menu if you have more than one fixed disk attached to your computer. If you choose this option, FDISK switches the current disk drive to the next fixed disk drive.

For example, if the current drive is drive C and if you choose option 5 on the FDISK Options menu, FDISK changes the current drive to drive D. You could then choose any of the first four FDISK options to prepare the second fixed disk for MS OS/2 *MULTIUSER*, or you could select option 5 once again to select the next drive. If there is not a third fixed disk, FDISK changes the current fixed drive from D back to C.

After you have selected the next drive, FDISK displays the FDISK options menu again. This time the information reported is for your second fixed disk.



# PART 3

## SYSTEM CONFIGURATION

# CHAPTER 4

## SYSTEM CONFIGURATION OVERVIEW

## INTRODUCTION

This section describes different ways you can configure your MS OS/2 *MULTIUSER* system and why one configuration may be better for you than another. Configuring an MS OS/2 *MULTIUSER* system involves creating users that can login to the system, setting up terminals attached to your system, and adding the required level of security to your system. More details on how to configure users, terminals, and security are contained in the following sections.

## SETTING UP THE FIRST USER PROGRAM

MS OS/2 *MULTIUSER* can be configured to start either the Program Selector, the MS OS/2 *MULTIUSER* command interpreter, or an application when a user logs into the system. The program that starts when a user logs into the system is called the "first user program." The first user program can be configured on a group basis or it can be different for every user. Refer to Chapter 5, "Configuring Profiles," for information on how to configure the first user program.

## PROGRAM SELECTOR VS COMMAND VS APPLICATION

The Program Selector allows users to customize a menu of programs where a user can choose the programs to start. Once a program is started, it is added to the Program Selector's task list. The task list is a list of programs currently running in the user's login. Selecting a program from the task list brings it to the foreground. A hotkey, called the Session Direct hotkey, is used to switch the Program Selector back to the foreground. When the Program Selector is in the foreground, the user can switch between the start programs menu and the task selector menu.

The string "C:\OS2\PSEL.EXE" (as the first user program) is used to start the Program Selector when the user logs in. Refer to "Program Selector" in Chapter 1 of the *Citrix MULTIUSER User's Guide* for more detailed information about running the Program Selector.

As the System Administrator, you may want users to start at the MS OS/2 *MULTIUSER* command prompt. From the MS OS/2 *MULTIUSER* command prompt, users can execute commands, utilities, or applications directly. It is also possible to get to an MS OS/2 *MULTIUSER* command prompt from the Program Selector by adding it as an option in the start programs menu. The string "C:\OS2\CMD.EXE" as the first user program is used to start the MS OS/2 *MULTIUSER* command interpreter when the user logs in. Refer to Chapter 3, "Running CMD," in the *Citrix MULTIUSER User's Guide* for more detailed information about running the MS OS/2 *MULTIUSER* command interpreter.

Users can also login directly to an application. To do this, enter the application's program name as the first user program. When the user exits the application that was started as the first user program, the user is automatically logged off. When a user is configured to login directly to an application, you may want to disable the hotkey used to create a session with the MS OS/2 *MULTIUSER* command prompt. The **ALT+TAB** hotkey, called the Session Create hotkey, can be disabled by leaving the Session Create hotkey field empty in the terminal profiles. Refer to Chapter 6, "Configuring Terminal Subsystems and Terminals."

## USING STARTUP FILES

There are several types of startup files, each with a different scope. These files are listed below.

■ *CONFIG.SYS*, the system configuration file

The *CONFIG.SYS* file is a configuration file and is required for the system to start. The commands within the file are processed when the system boots. There is only one *CONFIG.SYS* file in the system and it is located in the root directory of the startup drive. For example, if drive C is the startup drive, C:\CONFIG.SYS specifies the file.

■ *CONFIG.USR*, the configuration file for a *username*

The *CONFIG.USR* file is a configuration file and is optional. The commands within the file are processed when the user logs into the system and a new login is created. The *CONFIG.USR* file is located in the user's home directory. There may be a separate *CONFIG.USR* file for each user in the system. For example, if C:\USR\DAVIDH is the home directory for the *loginname* DAVIDH.WRKGRP, C:\USR\DAVIDH\CONFIG.USR specifies the file.

*CONFIG.USR* is useful for automatically starting programs and setting additional environment variables when a user logs into the system.

■ *STARTUP.CMD*, a startup batch file

The *STARTUP.CMD* file is a batch file and is optional. Different *STARTUP.CMD* files may be used for system startup and login-specific startup. Only one *STARTUP.CMD* file may be used for system startup. However, each user may have his own *STARTUP.CMD* file for login-specific startup. The system *STARTUP.CMD* file is useful for starting system-wide applications before users are allowed to login (for example, starting a network program). A user *STARTUP.CMD* file is useful for automatically starting user-specific applications.



To use a *STARTUP.CMD* file for system startup, put a *STARTUP.CMD* file in the root directory of the startup drive. The system uses the *loginname* and terminal specified in the system profile as the user and terminal where system startup commands run. After *CONFIG.SYS* handling, the system logs in this *loginname* at this terminal and invokes the command interpreter, CMD, to handle the *STARTUP.CMD* file. After the system *STARTUP.CMD* file has been processed, this login is disconnected from the terminal and terminals are initialized to allow users to login. For example, if drive C is the startup drive, then C:\STARTUP.CMD specifies the system startup file.

To use a *STARTUP.CMD* for user startup, put a *STARTUP.CMD* file in a user's working directory, which is the current directory when the user logs into the system, and configure the user's initial program as the Program Selector. When the Program Selector is started, the Program Selector invokes the command interpreter, CMD, to handle the *STARTUP.CMD* file. For example, if C:\USR\BRAD is the working directory for the *loginname* BRAD.MANAGER, C:\USR\BRAD\STARTUP.CMD specifies the user startup file.

■ **CMD /K [batch filename], a startup batch file option for CMD**

The /K option for CMD has a separate parameter that specifies the path and filename of a batch file that is executed when CMD starts. It is useful for setting up parameters for the session in which the command interpreter, CMD, executes.

## Configuration Files

Both the *CONFIG.SYS* and *CONFIGUSR* files use configuration commands. Any configuration command may be used in the *CONFIG.SYS* file. However, only a subset of configuration commands may be used in the *CONFIGUSR* file.

Configuration commands include the following:

AUTOFAIL	MEMMAN
BUFFERS	PAUSEONERROR
CALL	PRIORITY
CODEPAGE	PROTSHELL
COUNTRY	REBOOT
DEVICE	REM
DEVINFO	RESOURCE
DISKCACHE	RUN
HOSTNAME	SECURITY
IFS	SET
IOPL	SWAPPATH
LIBPATH	THREADS
LOG	TIMESLICE
MAINTENANCE	TRACE
MAXWAIT	TRACEBUF

The Configuration commands that may be used in the *CONFIGUSR* file are:

REM  
RUN  
SET

Refer to the *Citrix MULTIUSER Command Reference* for more information on these commands.

## Batch Files

The startup batch files, *STARTUP.CMD* and */K [batch file]*, may use any batch command or command. Batch files may not use configuration commands. Batch commands include the following:

CALL	IF
ECHO	PAUSE
ENDLOCAL	REM
EXTPROC	SET
FOR	SETLOCAL
GOTO	SHIFT

Refer to the *Citrix MULTIUSER Command Reference* for more information on these commands.

## MULTIUSER DIRECTORY STRUCTURE

MS OS/2 *MULTIUSER* introduces the concept of a home directory for each user. A user's home directory is created by MS OS/2 *MULTIUSER* when the System Administrator adds a user to the system through the CONFIG USER utility. For example, if the *username* DAVIDH is added, the home directory \USR\DAVIDH is created for *username* DAVIDH. The primary purpose of the home directory is to provide a place for the user's login-specific startup file, *CONFIG.USR*, to reside. Each user has his own unique home directory. The home directory applies to one and only one user.

Another new concept is that of a working directory for each user. Like the home directory, the working directory is defined for a user when that user is configured in the system. The purpose of the working directory is to define the directory that will be the user's current directory when the user logs into the system. The working directory is primarily

used to hold the user-specific startup file, *STARTUP.CMD*. By default, a user's working directory is the user's home directory. However, the working directory may be different from the home directory. In addition, the same working directory may be shared among several users. If the user's working directory is different from the home directory, it is the System Administrator's responsibility to ensure that the working directory exists.

In some cases, it is desirable to define the same working directory for more than one user. For example, there may be a set of users whose initial program is defined to be a specific application. When they log into the system, they will see only this application. In fact, their use of the system could be limited to the use of that single application. In this case, it might be appropriate to define the working directory for these users to be the same directory, most likely the directory where this application stores data.

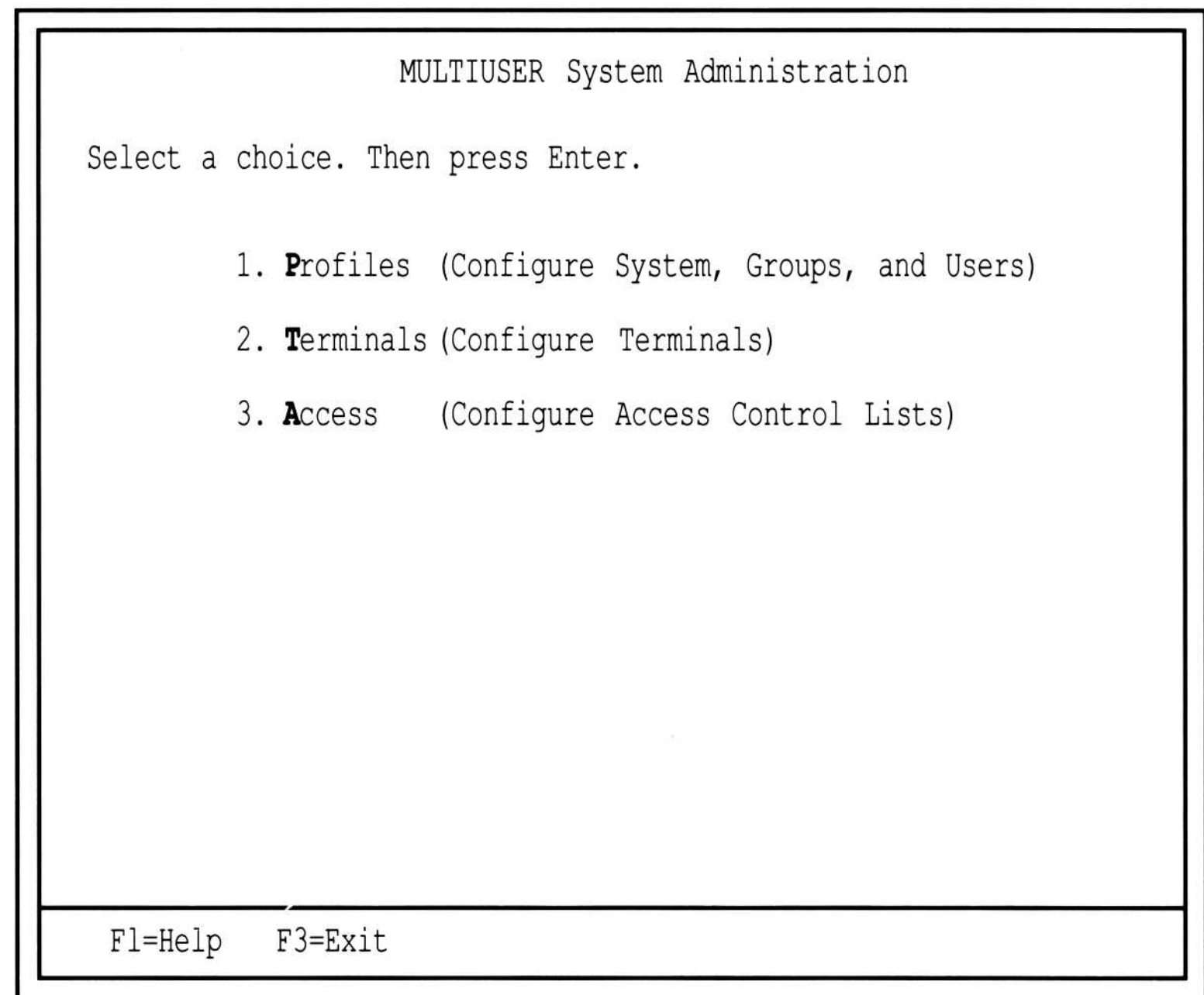
## INTRODUCTION TO THE CONFIG UTILITY

Configuration utilities allow users with Administrator class privilege to manage the various system and administration information. This includes the ability to create, modify, and delete users, groups, and terminals, as well as configure system parameters and administer security.

Profiles for the system, users, groups, and terminals are stored in files in predefined directories.

If no options are specified on the command line for the CONFIG command, a full screen menu will be displayed allowing any one of the configuration functions to be selected. The following figure shows how the menu will appear on the screen:





The following chapters describe configuring users, terminals, and security using the CONFIG utility.



# CHAPTER 5

## CONFIGURING PROFILES

## INTRODUCTION

This section describes how to manage the system, group, and user profiles. Profiles are used by the system to allocate system resources to users and define user-specific parameters such as the user's first program, security class level, working directory, and so forth. The command "CONFIG PROFILES" displays a menu with the system, group, and user profiles in an outline format. In this menu new groups and users can be added, renamed, or deleted. You can also edit any one of the profiles in the menu.

The Profiles option in the MULTIUSER System Administration menu is used to configure the system, group, and user profiles. Type CONFIG at the command prompt to get the MULTIUSER System Administration menu.

The following figure shows how the Profile Configuration menu appears:

Profile Configuration				
Profile	Options	More: ↓		
Profile	Dflt	Class Level	Last Login	Enabled
System Profile				
CITRIX (group)		Administrator		
└ CITRIX	*	Administrator	10/03/1990 08:40	Yes
UTILITIES (group)				
└ ANDYS		User	10/03/1990 10:25	Yes
└ CHRISL		User		Yes
└ KURTP		User		Yes
WRKGRP (group)				
└ ADMIN	*	Administrator	10/01/1990 11:27	Yes
└ ANDYS	*	User		Yes
└ ANNM	*	User		Yes
└ BRADP	*	User		Yes
└ CHRISL	*	User	09/28/1990 09:41	Yes
└ JOELS	*	User		Yes
└ KURTP	*	User		Yes
└ MIKED	*	User		Yes
F1=Help F3=Exit F5=Refresh F10=Actions				



## PROFILE NAMES

Profile names are used to identify groups, users and terminals. The names are from 1 to 15 characters long and can contain any character with an ASCII value greater than Hex 21 except the following:

< > ( ) [ ] { } . , : ; ' " \ \* ? % ^ & + =

Numbers and the following characters are not allowed as the first character of a name:

/ -

## RESOURCE MANAGEMENT

User resource limits can also be set when configuring user profiles. These limits are used to restrict/protect a user's consumption of the different types of resources.

These limits are enforced on a per-user basis and include all operations for the user. If a user logs in multiple times under the same *loginname*, the resources for the new login are added to the existing total from the first login. If the user logs out of the first login, all of those resources are freed up and the existing total decreased.

The QUERY LIMITS command can prove helpful in adjusting user limits. By having the users go about their normal day-to-day activities, the System Administrator can use QUERY LIMITS to get an idea of the amount of resources needed for each user.

If RESOURCE=OFF in *CONFIG.SYS*, these limits are not checked by the system.

## **Maximum Resource Limit**

The Maximum Resource Limit is the maximum amount of resources that a user can allocate. Any user attempt to allocate a resource past its limit results in an hard error popup on the user's screen. Only a user with sufficient (Administrator/Operator access) privilege can increase this value. If the limit is increased before the user replies to the popup, the user can retry the operation; otherwise, the only recourse for the user is to return the error to the application.

## **Minimum Resource Limit**

The Minimum Resource Limit is used to guarantee resource availability for each user once logged into the system. It reserves the amount of resources for the user, allowing no other user to allocate resources that would infringe on the user's minimum limit. As an example, user DAVIDH could have a minimum limit of 1 megabyte (MB) of virtual memory. If DAVIDH uses only 512K of his 1 MB minimum limit, there is at least 512K of free memory in the system while DAVIDH is logged in. Other users will have memory errors on memory allocations (assuming they have already attained their minimum limit) before his 512K of virtual memory is used.

Minimum resource limits are only in effect when the user is logged on. For example, if a user's minimum memory limit is set to 1 MB and the physical or virtual memory is not available, the user receives an error message and the login fails.

## SYSTEM PROFILE

The system profile contains default information used when creating new terminal and group profiles, as well as audit flags, event logging, and password rules. The system profile is created during MS OS/2 *MULTIUSER* installation with a set of predefined default settings. These settings can be modified by editing the system profile. To edit the system profile, enter the command "CONFIG PROFILES" and choose the "SYSTEM PROFILE" (topmost choice) for editing. You can also go directly to the system profile edit panel by entering the command "CONFIG SYSTEM" at the command prompt. The following figure shows how the system profile will appear in full screen window:

Edit System Profile		More: ↓
Profile	Options	
Terminal timeout .....	[60 ] minutes	
Default first user program .....	[C:\OS2\PSEL.EXE ]	
Default autologin user:		
Username .....	[ ]	
Groupname .....	[ ]	
Password .....	[ ]	
Options .....	[ ]	
Default user/terminal for system startup commands:		
Username .....	[ADMIN ]	
Groupname .....	[WRKGRP ]	
Terminalname .....	[CONSOLE ]	
Default hotkeys:		
Login switch .....	[{ctrl}{tab} ]	
Session switch .....	[{alt}{esc} ]	
Session direct .....	[{ctrl}{esc} ]	
F1=Help F2=Save F3=Exit F4=File F10=Actions		

The system profile contains the following information:

Terminal timeout:

Defines the number of minutes that a login must be inactive before it is automatically disconnected. A login is inactive when all sessions in the login are waiting for input from the keyboard. Leaving this field blank (or setting it to zero) disables terminal timeout checking.

Default first user program:

Defines the first user program name that is placed in the group profile when a new group profile is created. The first user program is the program that starts when a user logs in. This program can also be specified on a per user basis in the user's profile. Changing this field will not have any effect on existing group profiles.

Default autologin user:

Defines the autologin user who is placed in the terminal profile when a new terminal profile is created. Autologin is used to configure a terminal to automatically login a user when the terminal is powered on. This is useful when you want terminals to come up to a specific application whenever they are powered on. To do this, the autologin user must be configured with the application you want to start as the first user program. Leave these fields blank if you want terminals to power on with the Login: prompt. The Options field for the autologin user can be filled with any options supported by the login program as discussed in the section "Autologin Parameters" in Chapter 6. Changing these fields will not have any effect on existing terminal profiles.

Default user/terminal for system startup commands:

Defines the user and terminal to be used when the system startup files (*CONFIG.SYS* and *STARTUP.CMD*)



in the root directory of the boot drive are executed. Refer to Chapter 4, "Using Startup Files," for more information.

#### Default hotkeys:

Defines the hotkeys placed in the terminal profile when a new terminal profile is created. Hotkeys are key combinations that, when pressed simultaneously, invoke a defined function. To disable a particular hotkey, leave the field blank. Changes to the hotkey fields will not have any effect on existing terminal profiles.

- Login Switch - Switches between logins on the same terminal.
- Session Switch - Switches between sessions within a user login.
- Session Direct - Switches directly to the session registered for this hotkey (usually the Program Selector).
- Session Create - Creates a new session running CMD and makes it the foreground session.
- Print Screen - Prints the contents of the screen to LPT1.
- Print Screen Toggle - Toggles the printing of the screen on and off. When printing is on, all text written to the screen is also written to LPT1.
- Restart Terminal - Resets the terminal to its initial power on state.
- Stop Program - Stops the program currently running.



### Password settings:

Defines the rules for user passwords. Passwords are case sensitive. For example, "chrisl" is not the same as "CHRISL."

- **Maximum life** - Maximum number of days a *password* is valid until it must be changed. If the *password* is not changed within this time, the user must change the *password* during login in order to log into the system. Leave this field blank if the *password* does not expire.
- **Days before warning** - Number of days before *password* expiration when the user starts receiving messages at login time warning that the *password* is about to expire.
- **Minimum length** - The minimum number of characters that must be contained in a *password*. Enter a 0 in this field if a user is not required to have a *password*.
- **Maximum length** - The maximum number of characters that can be contained in a *password*. A *password* cannot be longer than 15 characters.
- **Syntax** - Defines the syntax rules that *passwords* must conform to. Enter a rule for each *password* character up to the maximum *password* length defined in the previous field. If no syntax is defined, the default is alphanumeric (R). The following characters are used to define the password syntax:

A - character must be an alpha character (A - Z).  
N - character must be a numeric character (0 - 9).  
C - consonant  
V - vowel  
R - character must be either alpha or numeric.  
X - any character is valid.

Event log flags:

Define the events that are to be logged when they occur. Refer to "Audit Events" in Chapter 7 for more information.

Audit flags:

Define the type of audit events that will be logged if the Audit field under Event logging in the previous field is enabled. Refer to "Audit Events" in Chapter 7 for more information.

Default resource limits:

Define the resource limits placed in the group profile when a new group profile is created. The resource limits define the maximum amount of a resource that a user can have at any given time as well as the minimum amount that must be available to the user in order to login. Changing these fields will not have any effect on existing group profiles.

These limits are initially set to defaults that should be good for most environments. The System Administrator can change these defaults if they are not right for this particular system.

The system maximum capacities for each resource are listed below:

Logins	No limit
Sessions	No limit
File handles	64,000
Virtual memory	Limited by physical memory size and disk space
System semaphores	256
Threads	Default is 512; can be up to 1024

The following resources are tracked by resource management.

- Logins - Maximum number of times that a user can be logged in at any given time.
- Sessions - Maximum number of sessions that a user can have at any given time.
- File handles - Maximum number of files that a user can have opened at any given time. Also, the minimum number of file handles that must be available to allocate to the user before logging in.
- Virtual memory - Maximum amount of virtual memory, in kilobytes, that a user can use at any given time. Also, the minimum amount of virtual memory that must be available to allocate to the user before logging in.
- System semaphores - Maximum number of system semaphores that a user can have open at any given time. Also, the minimum number of system semaphores that must be available to allocate to the user before logging in.

- Threads - Maximum number of threads that a user can have running at any given time. Also, the minimum number of threads that must be available to allocate to the user before logging in.

## GROUP PROFILES

Group profiles contain information used to manage the users who are members of the group. When a group profile is created, default information from the system profile is used to initialize the group profile settings. These settings can be modified by editing the group profile after it is created. To edit the group profile, enter CONFIG PROFILES and choose the group profile you want to edit by moving the selection bar over the *groupname* and pressing ENTER. You can also go directly to the group profile edit panel by entering the command CONFIG GROUP *groupname* at the command prompt.

Users who are members of the group will default to the settings in the group profile when they login under the *groupname*. Settings in the group profile can be overridden on a per-user basis in the user's profile. Refer to the section on User Profile for more information on editing user profiles.

Group profiles contain the following information:

### Audit flags:

Define the type of audit events that will be logged if the Audit field under Event logging in the System Profile is enabled. Refer to "Audit Events" in Chapter 7 for more information.

### Working directory:

Defines the directory that the user starts in during log in. Typically, this field is left blank so the user starts in his/her home directory.



First user program:

Defines the first user program name that automatically starts when the user logs in.

The first user program should be specified as a fully qualified drive and *pathname* and must have a *.EXE* or *.COM* extension. For example, if the Program Selector is the first user program, C:\OS2\PSEL.EXE would be entered.

Class level:

Defines the security class level of member users. Refer to Chapter 7, "Configuring Security," for more information on security classes.

Resource limits:

Define the resource limits that users default to when they login. The resource limits define the maximum amount of a resource that a user can have at any given time, as well as the minimum amount that must be available to the user in order to login.

Login time restrictions:

Define time spans for each day of the week when member users are permitted to login. Valid times are from 00:00 to 24:00. Any day that is left blank indicates that the user can login anytime on that day. Setting the Start and End times to be the same on a given day indicates that member users are not permitted to login at all on that day. A time span can cross over into the next day by entering an end time that is less than the start time. For example, if the login time for Monday is entered as 17:00 until 6:00, the user is permitted to login from 5:00 pm on Monday through 6:00 am on Tuesday.



## Creating a New Group

The following procedure explains how to create a new group. You can also create a new group by copying an existing group or by using the CONFIG GROUP command line options.

To create a new group:

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the SYSTEM PROFILE choice (topmost choice).
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose New.

The following window appears:

Create a New Group	
New groupname .....	[LDS ]
Esc=Cancel    F1=Help	

5. Type the name of the new group and press **ENTER**.

The new group is created with no members. See the section on adding users to a group to add members to the new group.

## Copying a Group

The following procedure explains how to copy a group to a new group. The settings from the original group are copied into the new group but no user memberships are copied.

To copy a group:

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *groupname* you want to copy.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Copy.

The following window appears:

Copy a Group	
Original groupname .....	:UTILITIES
New groupname .....	[ ]
Esc=Cancel      F1=Help	

5. Type the name of the new group and press **ENTER**.

The new group is created with no members. See the section on adding users to a group to add members to the new group.

## Deleting a Group

The following procedure explains how to delete a group. You can also delete a group using the CONFIG GROUP command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *groupname* that you want to delete.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Delete.

A message is displayed warning you that the group is about to be deleted.

5. Press **ENTER** to continue and delete the group.

All group memberships are also deleted but the users who are members of the group are not deleted.

## Modifying a Group

The following procedure explains how to modify a group profile. You can also modify a group profile using the CONFIG GROUP command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the

Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.

2. Move the selection bar over the *groupname* that you want to modify. You can press **ENTER** when the selection bar is over the *groupname* to edit the group profile or follow steps 3 and 4.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Edit. The following window appears:

Edit Group Profile		
Profile	Options	More: ↓
Groupname ..... :UTILITES		
Audit flags:		
Read .....	( ) Grant	( ) Deny
Write .....	( ) Grant	( ) Deny
Create .....	( ) Grant	( ) Deny
Execute .....	( ) Grant	(x) Deny
Delete .....	( ) Grant	( ) Deny
Attribute .....	( ) Grant	( ) Deny
Login .....	(x) Grant	(x) Deny
Use .....	( ) Grant	( ) Deny
Query self .....	( ) Grant	( ) Deny
Query other .....	( ) Grant	(x) Deny
Modify self .....	( ) Grant	( ) Deny
Modify other .....	( ) Grant	(x) Deny
Working directory .....	[	]
First user program .....	[C:\OS2\PSEL.EXE	]
F1=Help   F2=Save   F3=Exit   F4=File   F10=Actions		

5. Make the changes you want to the group profile, then press **F4** to save the changes and exit. You can press **F3** to exit without saving changes.

## Renaming a Group

The following procedure explains how to rename a group. You can also rename a group using the CONFIG GROUP command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *groupname* that you want to rename.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Rename.

The following window appears:

Rename Group	
Old groupname .....	:UTILITIES
New groupname .....	[                      ]
Esc=Cancel      F1=Help	

5. Type the new name of the group and press **ENTER**.

The group is renamed.



## USER PROFILES

User profiles contain information used to manage the users and group memberships. Information in a user profile consists of user data and information for each group that the user is a member of with settings that override the settings defined in the group profiles. The user also has a default group membership that is used when the *groupname* is not specified with the *username*. To edit a user profile, enter the command CONFIG PROFILES and choose the user profile under the group membership you want to edit by moving the selection bar over the *username* and pressing enter. You can also go directly to the user profile edit panel by entering the command "CONFIG USER *username*[.*groupname*]" at the command prompt.

User profiles contain the following information:

Default *groupname*:

Defines the default group membership for the user. The default group for a user can be changed in the Profiles Configuration window.

Full name:

Formal name of the user.

Comment:

Free form area used to hold information about a user.

Account Expiration:

Defines the date and time when the user account will expire. When a user account expires, the user can no longer login to the system. Leaving this field blank indicates that the user account does not have an expiration date. The date/time must be entered as mm-dd-yyyy hh:mm.

**User state:**

Defines the state of the group membership (*loginname*). If the state is set to disabled, the user cannot login under the group membership.

**Audit flags:**

Define the type of Audit events that will be logged if the Audit field under Event logging in the System Profile is enabled. Refer to "Audit Events" in Chapter 7 for more information.

**Auto disconnect login:**

Specifies whether the user's login should be disconnected when the terminal connection is lost or the login times out. If auto disconnect is enabled, the user's login will be disconnected in the above situations and will be recovered the next time the user logs in. If auto disconnect is not enabled, the login will be terminated.

**Auto connect login:**

Specifies how a user's disconnected login is reconnected. If auto connect at any terminal is selected, all disconnected logins for the user will be reconnected when the user logs in to the system. If auto connect at the same terminal only is selected, only disconnected logins for the user that were previously connected to the same terminal that the user is logging into will be connected.

**Working directory:**

Specifies the directory that the user starts in when the user logs in. Typically, this field is left blank so the user starts in his/her home directory. If this field is blank and a working directory is specified in the group profile, the user will login to the working directory specified in the group profile.

First user program:

Defines the first user program name that starts when the user logs in. If this field is left blank, the first user program defined in the group profile is used.

Class level:

Defines the security class level of the user. Refer to Chapter 7, "Configuring Security" for more information on security classes.

Resource limits:

Defines the resource limits for the user. The resource limits define the maximum amount of a resource that a user can have at any given time as well as the minimum amount that must be available to the user in order to login. Limits that are left blank will default to the values defined in the group profile.

Login time restrictions:

Define time spans for each day of the week when member users are permitted to login. Valid times are from 00:00 to 24:00. Any day that is left blank indicates that the user can login anytime on that day. Setting the Start and End times to be the same on a given day indicates that member users are not permitted to login at all on that day. A time span can cross over into the next day by entering an end time that is less than the start time. For example, if the login time for Monday is entered as 17:00 until 6:00, the user is permitted to login from 5:00 pm on Monday through 6:00 am on Tuesday.

## Creating a New User

The following procedure explains how to create a new user. You can also create a user by copying an existing user or by using the CONFIG USER command line options.

To create a new user:

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the groupname that you want the user to be a member of or over one of the users who is already a member of the group.
3. Press the **F10** (or **ALT**) key to go to the action bar.
4. Choose Profile from the action bar, then choose New.

The following window appears:

Create a New User	
Group membership .....	:WRKGRP
New username .....	[MIKEH ]
Drive of home directory .....	[C]
Esc=Cancel    F1=Help	

5. Type the name of the new user and press **ENTER**.

The new user is created and added as a member to the group that was highlighted with the selection bar.



## Copying a User

The following procedure explains how to copy a user to a new user. The settings from the original user are copied into the new user. However, the new user is only a member of the group that the selection bar was under when the original user was selected.

To copy a user:

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *username* under the group membership that you want to copy.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Copy.

The following window appears:

Copy a User	
Original username .....	:CHRISL
New username .....	[                      ]
Esc=Cancel      F1=Help	

5. Type the name of the new user and press **ENTER**.



The new user is created as a member of the group where the selected user is a member.

## Deleting a User

The following procedure explains how to delete a user. A user can also be deleted using the CONFIG USER command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line.
2. Press **F10** (or **ALT**) to go to the action bar.
3. Choose Options from the action bar, then choose Sort by user.

The profile is sorted by *usernames* with the *groupnames* that the user is a member of indented under each *username*.

4. Move the selection bar over the username that you want to delete.
5. Press **F10** (or **ALT**) to go to the action bar.
6. Choose Profile from the action bar, then choose Delete.

A message is displayed warning you that the user is about to be deleted.

7. Press **ENTER** to continue and delete the user.

The user is deleted.

## Modifying a User

The following procedure explains how to modify a user profile and the group membership information of the user. A user profile can also be modified using the CONFIG USER command line options.

To modify a user:

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *username* within the group that you want to modify. You can press **ENTER** when the selection bar is over the username to edit the user profile, or follow steps 3 and 4.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Edit. The following window appears:

Edit User Profile		More: ↓
Profile	Options	
Username .....	:CHRISL	
Default groupname .....	:WRKGRP	
Full name .....	[C.T. Lehman ]	
Comments .....	[Utilities Developer ]	
User Comments .....	[ ]	
Account expiration .....	[09-22-1991 00:00] (mm-dd-yyyy hh:mm)	
Group membership settings:		
Groupname .....	:UTILITIES	
User state .....	(x) Enabled ( ) Disabled	
Audit flags:		
Read .....	( ) Grant ( ) Deny	
Write .....	( ) Grant ( ) Deny	
Create .....	( ) Grant ( ) Deny	
Execute .....	( ) Grant ( ) Deny	
F1=Help F2=Save F3=Exit F4=File F10=Actions		

5. Make the changes you want to the user profile and group membership, then press **F4** to save the changes and exit. You can press **F3** to exit without saving changes.

## Renaming a User

The following procedure explains how to rename a user. A user can also be renamed using the CONFIG USER command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *username* that you want to rename.

3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Rename.

The following window appears:

Rename User	
Old username .....	:CHRISL
New username .....	[                      ]
Esc=Cancel    F1=Help	

5. Type the new name of the user and press **ENTER**.

The user is renamed.

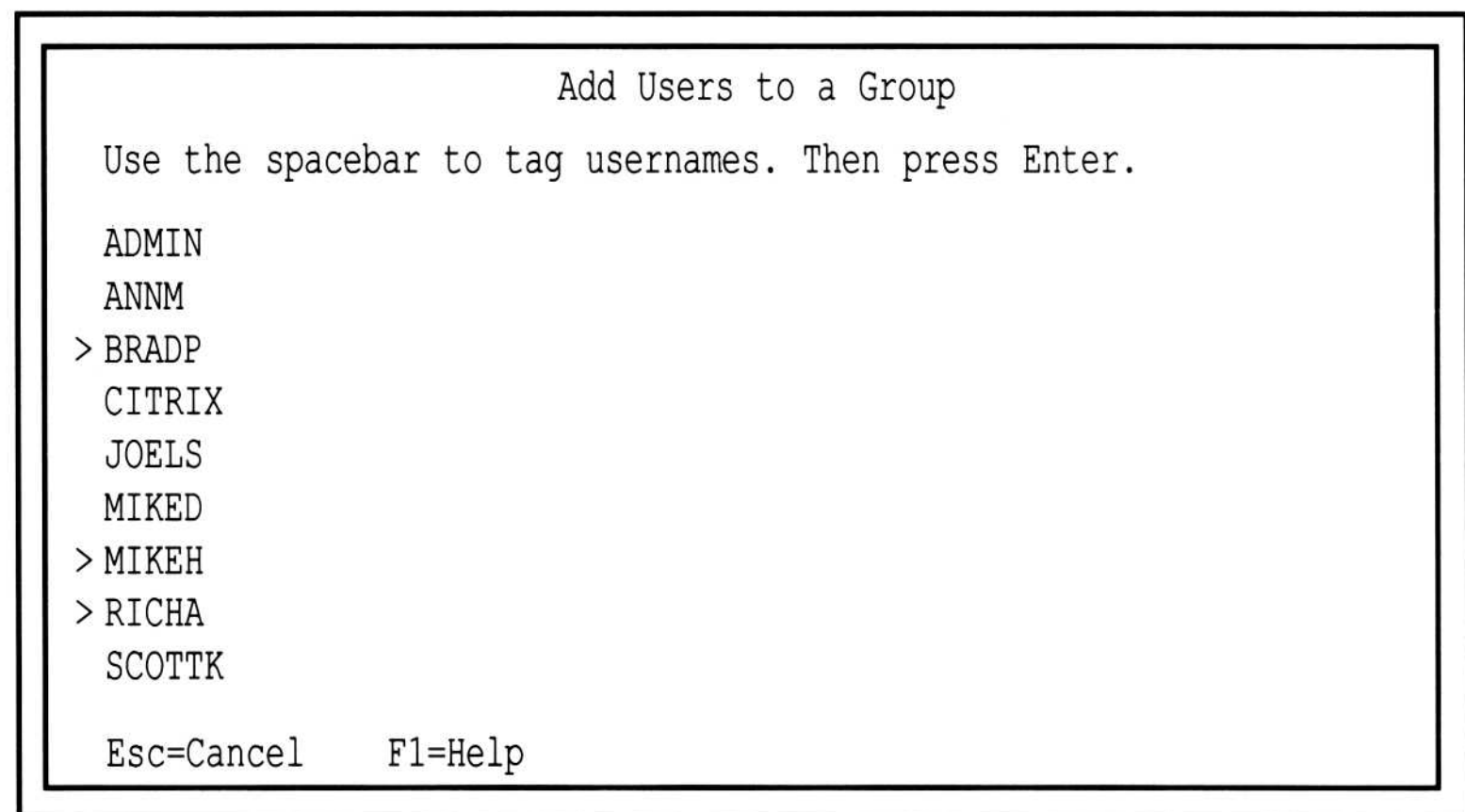
## Adding Users to a Group

The following procedure explains how to add users as members to a group. A user can also be added as a member to a group by using the CONFIG USER command line options.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *groupname* that you want to add users as members to or over one of the users who is already a member of the group.

3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Add members.

The following window displays the users who are not members of the group appears:



5. Use the **SPACEBAR** to tag the users that you want to add as members. To remove the tag from a user, press the **SPACEBAR** again. When all users that you want to add as members to the group are tagged, press **ENTER**.

The tagged users are added as members to the group.

## Deleting a User From a Group

The following procedure explains how to delete a user's membership to a group. A user's membership to a group can also be deleted by using the CONFIG USER command line options.



1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *username* under the group where you want to delete the user's membership.
3. Press **F10** (or **ALT**) to go to the action bar.
4. Choose Profile from the action bar, then choose Delete.

If this is the only group that the user is a member of, a message is displayed warning you that the user is about to be deleted. If the user is also a member of other groups, a message is displayed warning that the membership to this group is about to be deleted.

5. Press **ENTER** to continue and delete the user or user membership.

If this was the only group that the user was a member of, the user is deleted. If the user is a member of other groups, only the user membership to this group is deleted.

## **Changing a User's Password**

The following procedure explains how to change a user's password. You can also change a user's password by using the PASSWORD utility.

1. Display the Profile Configuration menu by entering the command CONFIG PROFILES at the command line. If the Profile Configuration menu is already displayed, make sure the profiles are sorted by groups.
2. Move the selection bar over the *username* within the group that you want to change the password for.
3. Press ENTER to edit the user profile. The Edit User Profile menu appears.
4. Press **F10** (or **ALT**) to go to the action bar.
5. Choose Options from the action bar, then choose Change password. The following window appears:

Change Password	
User name .....	:CHRISL
Groupname .....	:WRKGRP
Current password .....	[ ]
New password .....	[ ]
Esc=Cancel    F1=Help	

6. Type the Current password and the New password in the appropriate fields. The cursor will move but no characters will be displayed. If you are an Administrator class and the current password was forgotten, do not type anything in the Current password field.
7. Press ENTER. The following window appears:

Confirm New Password
Re-enter New password ..... [                      ]
Esc=Cancel      F1=Help

8. Type in the new password again and press ENTER.

The password is changed.

# CHAPTER 6

## CONFIGURING TERMINAL SUBSYSTEMS AND TERMINALS

## TERMINOLOGY

Terms used in this chapter are defined below.

*Adapter Card.* An adapter card is a hardware device that is inserted into one of the expansion slots of the host computer to extend or expand the capabilities of the host computer. The pertinent adapter cards discussed in this section are multi-port serial port adapter cards.

*Base MS OS/2 MULTIUSER Asynchronous Communications Device Driver.* The Base MS OS/2 MULTIUSER Asynchronous Communications Device Driver (Async Device Driver) is a program used so that multiple serial ports with differing hardware specifics can run the same serial communications software. The Base Async Device Driver assumes base functions and local functions as defined by IBM's *I/O Subsystems and Device Support Manual*. MS OS/2 MULTIUSER relies on the existence of an Async Device Driver for any serial port terminal subsystems. The serial ports on the host computer are supported by the async device drivers named COM01.SYS for Industry Standard Architecture computers like COMPAQ's and COM02.SYS for Micro Channel computers like IBM's.

*Baud Rate.* This is the rate at which data flows across a serial cable in bits per second.

*Cable.* A cable is a wire that connects the terminal to the computer. Keyboard data sent from the terminal travels across the cable to the host computer. Video data sent from the host computer travels across the cable to the terminal.

*Data Bits.* In a serial data stream, this number indicates the number of bits that make up a word or unit of data. Typically, this value is 8 bits for standard ASCII.



*Host Computer.* The host computer is the computer on which MS OS/2 *MULTIUSER* is running. The host computer typically contains a fixed disk, one or more floppy diskette drives, memory, a display, and a keyboard.

*Hotkey.* A hotkey is a key stroke or combination of key strokes that, when pressed, cause the operating system (as opposed to an application program) to perform a function. For example, to switch between one application and another application, press **ALT+ESC**. This combination of key strokes is known as a hotkey.

*Parity.* This value is a bit inserted between words in a serial data stream primarily used for determining the integrity of the data.

*Serial Port.* A serial port, or asynchronous communications port, is hardware that allows for transmitting and receiving data between the host computer and the terminal. Terminal subsystems contain one or several serial ports. Most host computers contain at least one serial port.

*Stop Bits.* This is the number of bits inserted between words in a serial data stream to indicate the end of one word and the beginning of another word.

*System Console.* The host computer's display and keyboard is considered a unique terminal, called the system console.

*Terminal.* A terminal is a combination of a computer display and a keyboard. The terminal's keyboard is the primary means by which you enter commands and keyboard data. Results of those commands and display data are shown on the terminal's display. MS OS/2 *MULTIUSER* supports a variety of terminals that attach to the computer's serial port (refer to Appendix D, "Setting Up Specific Terminal Types").

*Terminal Subsystem.* A terminal subsystem is the hardware in the host computer that transports the data to and from the terminal. There are two types of terminal subsystems in MS OS/2 *MULTIUSER*; the system console terminal subsystem and the serial port terminal subsystem.

## BACKGROUND ON ADDING A TERMINAL SUBSYSTEM

In order to support multiple users, terminals must be attached to the host computer. This entails making a number of decisions before proceeding. The first decision you must make is how many terminals you want to have using the system at any given time.

If the number of users is only two or three, no additional hardware may be required beyond the host computer. This depends on the number of serial ports existing on the host computer. Typically, the host system contains one or two serial ports that are supported by the async device driver. Each of these serial ports can have a terminal attached. This means that most computers can support two or three users without additional hardware.

However, if the number of terminals needed is four or greater, you will need to install a multi-port async adapter. These come in many shapes and sizes, and are detailed in appendix C, "Installing Async Ports for MS OS/2 *MULTIUSER*." Typically, these serial port subsystems allow for incremental additions of serial ports in increments of 2, 4, 8, and 16.

## TERMINAL SUBSYSTEMS

### Installing an Adapter

Two steps are required to install an adapter card: hardware installation and software installation.

#### Hardware Installation

The adapter card must be physically configured according to the requirements of the manufacturer. This entails physically setting the switches on the adapter card before inserting the adapter card into the expansion slots of the host computer. For Micro Channel systems, the configuration switches are stored in nonvolatile memory on the host computer and must be set by running the computer manufacturer's diagnostics and setup diskette. Each adapter manufacturer has a different set of requirements that must be met for each of his adapter cards. For further information, refer to Appendix C, "Installing Async Ports for MS OS/2 *MULTIUSER*." If there is no information in the appendix for the adapter card in question, consult the manufacturer's publications for specific hardware installation instructions.

#### Software Installation

Some non-intelligent serial adapter cards are supported with the Async Device Driver software provided in MS OS/2 *MULTIUSER*, meaning no software installation of the device driver is required. These adapter cards have either one or two serial ports and are supported by the same device driver that supports the serial ports on the host computer. The manufacturer's documentation should provide the information on whether a new device driver is required or if the MS OS/2 *MULTIUSER* Async Device Driver suffices. Usually if the board has more than two serial ports on it, it probably requires the installation of a manufacturer-unique device driver.



If a device driver is required, the device driver must be installed to allow for the adapter to be used. This entails modifying the *CONFIG.SYS* file by adding a "DEVICE =" statement for the adapter card. Some manufacturers provide installation programs to do this modification. Others require that the *CONFIG.SYS* file be edited using an editor. Either way, the device becomes operational only after the *CONFIG.SYS* file is modified and the system is rebooted. Refer to Appendix C, "Installing Async Ports for MS OS/2 *MULTIUSER*" for further information. If there is no information in the appendix for the adapter card in question, consult the manufacturer's publications for specific software installation instructions.

## Adding a Terminal Subsystem

MS OS/2 *MULTIUSER* provides a predefined terminal subsystem configuration file for the MS OS/2 Async Device Driver. Appendix C, "Installing Async Ports for MS OS/2 *MULTIUSER*" lists those adapter cards that are supported by the MS OS/2 Async Device Driver Terminal Subsystem. If the adapter card is not listed in the appendix, but it has an MS OS/2 *MULTIUSER* async device driver, that adapter can usually be installed and run with MS OS/2 *MULTIUSER*.

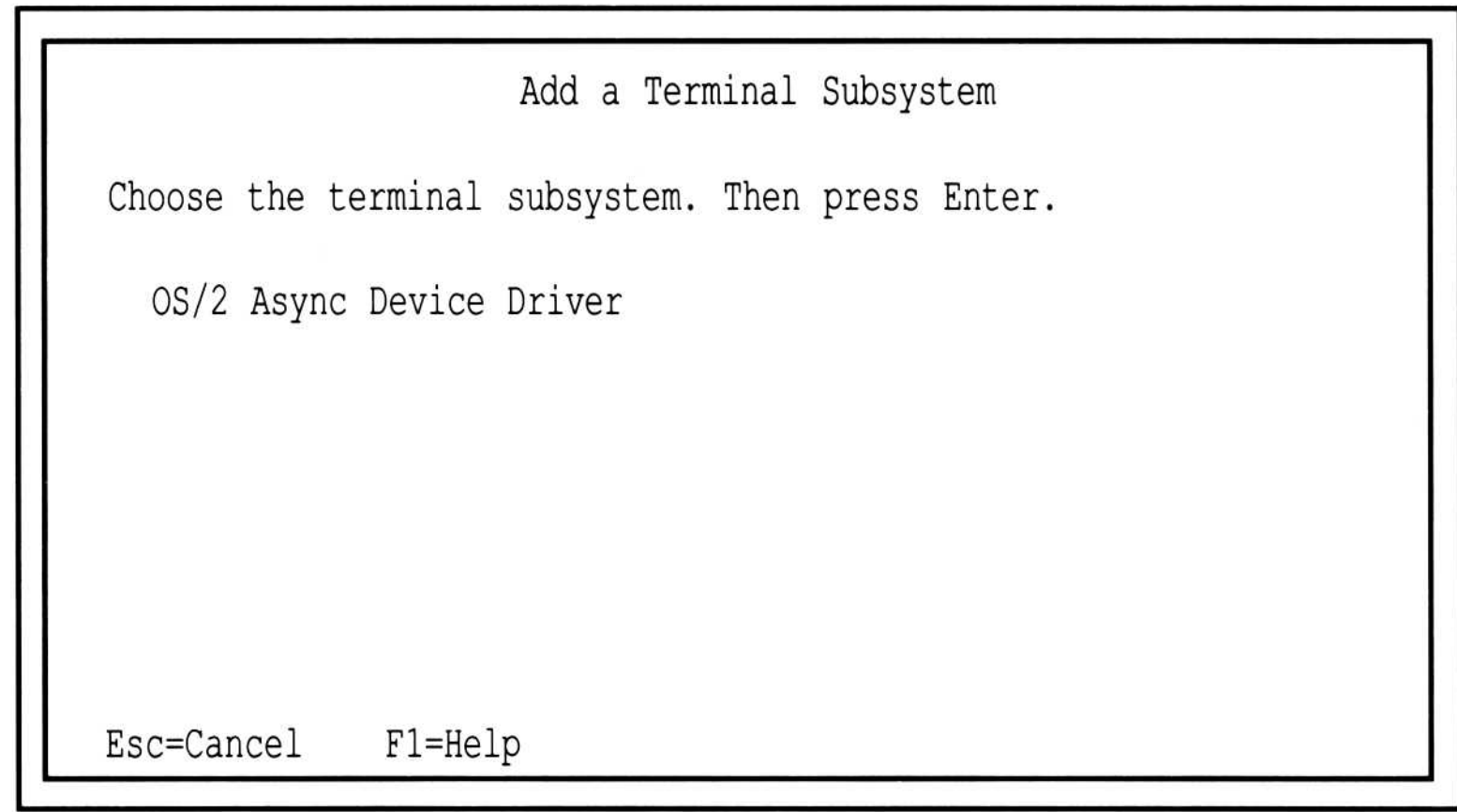
To add a terminal subsystem, use the following steps:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line. The following panel appears:

Terminal Configuration			
Profile	Options		
Terminal	Terminal Type	Baud	Device
System Profile			
Console			
└─ CONSOLE	CONSOLE	0	CON
F1=Help    F3=Exit    F5=Refresh    F10=Actions			

- Move the selection bar to "SYSTEM PROFILE" (topmost choice).
- Press **F10** to go to the action bar, select PROFILE, then select ADD Subsystem. The following window appears:





4. Move the selection bar to the terminal subsystem that is to be added and press **ENTER**. If the terminal subsystem does not exist in the list, it must be created. See "Creating a New Terminal Subsystem."

## Deleting a Terminal Subsystem

To delete a terminal subsystem, follow the steps below:

1. Display the terminal configuration panel by entering the command **CONFIG TERMINAL** at the command line.
2. Move the selection bar to the terminal subsystem that is to be deleted.
3. Press **F10** to go to the action bar, select Profile, then select Delete.
4. When the warning message appears, press **ENTER**.

## CONFIGURING TERMINALS, CABLES, AND SERIAL PORTS

All terminals with the exception of the system console must be attached to a serial port on the host computer. There are three steps to this process:

1. Setup the terminal and serial port.
2. Add the terminal to the system configuration.
3. Attach the terminal to the system with a serial cable.

The steps to add/delete a terminal subsystem are described in the previous section. Other instructions follow.

### Setting up Your Terminal

MS OS/2 *MULTIUSER* provides support for many manufacturers' serial terminals. All of the terminals MS OS/2 *MULTIUSER* supports are capable of providing what is known as "personal computer scan codes." This means the terminals have the ability to nearly exactly mimic the behavior of a personal computer keyboard across a serial port. This feature is required to provide for the highest degree of compatibility with existing MS OS/2 *MULTIUSER* text based applications.

After choosing from the list of supported terminals in Appendix D, "Setting Up Specific Terminal Types," it is necessary to configure the terminal using the terminal manufacturer's "setup" program. The details of how to run the terminal manufacturer's "setup" program are provided in the appendix. Further information can be found in the publications provided with the terminal. Also check your README files for possible new terminal types not in this manual.

When terminals arrive from the factory, they have certain default settings. Your installation will probably not use the terminal in exactly the same way as it is set up at the factory, so you will have to configure the terminal according to your specific requirements.

Some of these requirements (such as keyboard character set and terminal emulation) are fixed requirements due to the nature of the terminal and MS OS/2 *MULTIUSER*. Some requirements (such as baud rate) are determined by the configuration of the serial ports on your system and some parameters (such as keyboard key click on/off) are up to the user. Note that the link between the terminal and the host computer is such that certain parameters (such as baud rate) must be correct on both sides of the link or there can be no communication.

## Creating A New Terminal Profile

To add a terminal to an existing MS OS/2 *MULTIUSER* system, the terminal type, the serial port location, and method of cabling must be identified. The terminal must be cabled to a serial port on the host computer and the required parameters must be set up using the terminal manufacturer's setup program. The baud rate of the terminal must be the same as the baud rate of the serial port.

To create a terminal profile, follow the steps below:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line. A sample full screen panel appears:

Terminal Configuration			
Profile	Options		
Terminal	Terminal Type	Baud	Device
System Profile			
Console			
└─ CONSOLE	CONSOLE	0	CON
OS\2 Async Device Driver			
└─ SERIAL0	MULTIUSER Link	19200	COM1
└─ SERIAL1	ADDS 2025	19200	COM2
└─ SERAIL2	IBM 3151	9600	COM3
└─ SERIAL3	TeleVideo 965	9600	COM4
F1=Help  F3=Exit  F5=Refresh  F10=Actions			

2. Move the selection bar to the line with the terminal subsystem on which the terminal is attached.

As stated above, some terminal subsystem configurations are predefined. If your subsystem matches one in the list of subsystems, that subsystem should be chosen. If, however, your terminal subsystem has a supporting MS OS/2 *MULTIUSER* Async device driver, refer to "Creating a New Terminal Subsystem."

3. Press **F10** to go to the action bar, select Profile, then select New. The following panel appears:

Create a New Terminal	
Terminal subsystem .....	:OS/2 Async Device Driver
New terminalname .....	[                      ]
Esc=Cancel      F1=Help	

4. Type the terminal name you want to use and press **ENTER**.

Any text string will suffice for the terminal name, but it is important that this be remembered. It should be chosen to uniquely identify the terminal and/or the terminal's port location. The terminal name is used as an input parameter and sometimes as an output parameter on many of the MS OS/2 *MULTIUSER* utilities. It is recommended that a root name like "TERM" with a relative number extension be used; for example, TERM01, TERM02, and so forth.

5. Move the selection bar to the device name that is to be selected and press **ENTER**.

The device name addresses the port on the terminal subsystem to which the terminal is cabled. This means you choose the device driver name or serial port address associated with the serial port connecting the terminal to the host computer. For example, if the terminal is cabled to the host computer serial port, choose the device name COM1.

6. Move the selection bar to the selected terminal type and press **ENTER**. The system automatically chooses the default parameters for the terminal type. The "Edit



Terminal Profile" screen will then be displayed. Make any required changes to the terminal profile at this point. Follow the procedures listed in the "Modifying a Terminal Profile" section. When the edit screen is saved and exited, the system will attempt to initialize the terminal.

## Modifying a Terminal Profile

Follow the steps below to modify a terminal profile:

1. Display the terminal configuration panel by entering the command CONFIG TERMINAL at the command line.
2. Move the selection bar over the *terminalname* that is to be edited. Alternately, the *terminalname* may be included as a parameter on the CONFIG TERMINAL command.
3. Press **F10** to go to the action bar, select Profile, then select Edit. The following panel appears:

Edit Terminal Profile		More:	↓
Profile	Options		
Terminalname .....	SERIAL1		
Terminal subsystem .....	OS/2 Async Device Driver		
Terminal type .....	ADDS 2025		
Device name .....	COM2		
Comments .....	[ ]		
Autologin user:			
Username .....	[ ]		
Groupname .....	[ ]		
Password .....	[ ]		
Options .....	[ ]		
Initial terminal state .....	(x) Enabled ( ) Disabled		
Hotkeys:			
Login switch .....	[{ctrl}{tab}]		
Session switch .....	[{alt}{esc}]		
F1=Help F2=Save F3=Exit F4=File F10=Actions			

4. Edit the appropriate terminal configuration parameters.
5. Press F2 to save the parameters in the profile.

After the terminal configuration is modified, the system is signaled that a new terminal has been added or an old terminal's configuration has changed. The system will attempt to initialize the terminal or set the changed configuration parameter at the terminal depending on the parameters modified. The login program or the initial user program will be displayed when a connection is established. If the terminal is turned off or unplugged from the power source, powering up the terminal should bring the terminal up. If this fails to bring about the desired result, run RESET TERMINAL. If this fails to bring the terminal up, the configuration steps must be retraced to determine the cause of the failure. You may also refer to Chapter 16, "Problem Determination," for help in resolving terminal configuration problems.

## Terminal Parameters to Modify

### Terminal Type:

The terminal type is entered so that the default parameters for the terminal type can be automatically chosen. The list of terminals that are supported is documented in Appendix D, "Setting Up Specific Terminal Types."

### Autologin Parameters:

MS OS/2 *MULTIUSER* provides a secure system environment by requiring, at the System Administrator's discretion, that each user login to the system with a *username* and *password*. For system installations that are not concerned about breaches in login security, the System Administrator can bypass this by installing the autologin parameters. The *username.groupname* may be entered and an optional *password* may be entered using the /PASSWORD option. For

example, assume SCOTTK is a *username* in the group WRKGRP with the *password* SCOTTKPW. To bypass the login screen and proceed directly to the user default program, the System Administrator would enter the autologin parameters in the appropriate fields.

Parameters may also be passed to the login program by entering them in the Options field. The following options are supported:

/V

Puts the login program in verbose mode. This option causes the login program to display data that may be used for problem determination.

/MSG: *filename*

This option allows for the displaying of a file name *filename* during the login process prior to displaying the login prompt. This is used to transfer system-wide information to all users logging in to the system.

/HOST

This option allows for the displaying of the hostname with the login prompt. The hostname is defined via the HOSTNAME configuration command.

### Initial Terminal State:

MS OS/2 *MULTIUSER* allows a System Administrator to set the initial terminal state. The state of the terminal can be set so that the terminal is initially enabled or disabled. If the terminal is initially enabled, the terminal will automatically present the login screen or the initial user program screen at power up. If, however, the terminal is set to be initially disabled, the terminal will not present the login screen or the initial user program at power up. The terminal will stay in the disabled state until the RESET TERMINAL program is run from another terminal.

## Hotkeys:

MS OS/2 *MULTIUSER* permits a System Administrator to configure the hotkey key sequences for the terminal. The following hotkeys are configurable:

### Login Switch

Switches between logins on the same terminal. The default is **CTRL+TAB**.

### Session Switch

Switches between sessions within a user login. The default is **ALT+ESC**.

### Session Direct

Switches directly to the session registered for this hotkey (usually the Program Selector). The default is **CTRL+ESC**.

### Session Create

Creates a new session running CMD and makes it the foreground session. The default is **ALT+TAB**.

### Print Screen

Prints the contents of the screen to LPT1. The default is **PRINT SCREEN**.

### Print Screen Toggle

Toggles printing of the screen on and off. When printing is on, all text written to the screen is also written to LPT1. The default is **CTRL+PRINT SCREEN**.

### Stop Program

Stops the program currently running. The default is **CTRL+C** or **CTRL+Break**.

### Restart Terminal

Terminates all applications running on the terminal, then restarts the terminal. The default is **CTRL+ALT+DEL**.



Note that the Restart Terminal hotkey is not configurable on the console except when REBOOT=OFF in the system configuration file, *CONFIG.SYS*.

A hotkey is invoked when all keystrokes defined for the hotkey are pressed down at the same time.

A hotkey combination can consist of up to two shift keys and one non-shift key. The combination must be written by surrounding each key in brackets ({}). The lists below show how to denote shift keys and special non-shift keys. The following is an example of a hotkey combination that is invoked by pressing the "CTRL" key and the letter "A":

{ctrl}{a}

#### SHIFT KEYS

{alt}	Either <b>ALT</b> key
{ltalt}	Left <b>ALT</b> key
{rtalt}	Right <b>ALT</b> key
{ctrl}	Either <b>CTRL</b> key
{ltctrl}	Left <b>CTRL</b> key
{rtctrl}	Right <b>CTRL</b> key
{shift}	Either <b>SHIFT</b> key
{ltshift}	Left <b>SHIFT</b> key
{rtshift}	Right <b>SHIFT</b> key
{caps}	<b>CAPS LOCK</b> key
{num}	<b>NUM LOCK</b> key
{scroll}	<b>SCROLL LOCK</b> key
{sysreq}	<b>SYS REQUEST</b> key

#### SPECIAL NON-SHIFT KEYS

{esc}	<b>ESC</b> key
{enter}	<b>ENTER</b> key
{backspace}	<b>BACKSPACE</b> key
{tab}	<b>TAB</b> key



{up}	UP arrow key
{down}	DOWN arrow key
{right}	RIGHT arrow key
{left}	LEFT arrow key
{home}	HOME key
{end}	END key
{delete}	DELETE key
{insert}	INSERT key
{pageup}	PAGE UP key
{pagedown}	PAGE DOWN key
{pause}	PAUSE key
{prtscrn}	PRINT SCREEN key
{minus}	- on keypad
{plus}	+ on keypad
{star}	* on keypad
{f1}	F1 key, and so forth

**NOTE:** Some of the keys on the top row of the keyboard are not valid with the **CTRL** key.

## Host to Serial Terminal Communications

The communications subsystem provides many options for connecting terminals to the host system. The options will vary depending on the type of host communications connection and the type of terminal connection. Many of the parameters are determined by the type of host communication and terminal connection being used. Some typical options determined by system and terminal types are connector types, wire types, and signal interfaces. The options that can be selected via **CONFIG TERMINAL** are data rates, data protocols, data pacing, and terminal connect/disconnect signals. The characteristics of the selectable options are described in further detail below.

## Data Flow

One of the fundamental characteristics of communications in a multiuser environment is the transfer of data to and from the system and user. While this may seem basic, the following information may help in configuring a reliable, secure multiuser environment. An industry standard signal interface for making host/terminal connections is the RS-232 asynchronous interface. A common choice with this type of interface is the data rate or baud rate. While the highest data rate possible is usually the most desirable, it is also an option that quickly exceeds the limits of the interface or, sometimes, the system. It is important to be aware of the basic factors associated with data rates.

A data flow limitation that might be exceeded is baud rate versus distance. As the distance from the host system to the user terminal increases, the data signal is attenuated. As the data signal is reduced, it becomes vulnerable to data loss due to interface limitations and noise interference. Data rates versus distance values are not typically stated due to the number of significant electrical factors associated with the equipment and installation environment.

Recommended information of this sort may be obtained from the equipment's installation guide or where the equipment/supplies were purchased. A good starting point may be the defaults for the communications subsystem being configured. Often, if data is being lost, the problem can be solved by reducing the data rate via the configuration utilities.

Another data flow limitation that may be exceeded is baud rate versus system port type. For asynchronous communications, the system port type may be divided into two general categories: non-intelligent and intelligent. The non-intelligent port requires the host system to service the

port at a very low level, taking time from the host system's processing power. As the data rate is increased, the non-intelligent port requires more immediate response from the host system. Since multiple non-intelligent ports may be in the host system, the host system may lose data if the baud rates on these ports exceed the system capabilities.

Intelligent ports have an additional microprocessor that removes the requirement of immediate host system processing. The intelligent port collects data and provides blocks of data when the host system is ready to accept it. This interrupt overhead problem within the host system can be solved by reducing the data rate or baud rate with the CONFIG TERMINAL program. If data is lost with either the non-intelligent or intelligent port, it can often be corrected by reducing the baud rate.

## Data Format and Handshaking Techniques

The format of the data being transferred from the system and terminal has programmable characteristics. "Handshaking" is the term used to describe the method used to suppress data transmissions when a peripheral device or system cannot accept any more data due to processing or storage capabilities. While these two options are very different, one may impose limits on the other. The options for the data format are character size, parity mode, and stop bits.

### **Data Bits:**

The character size parameters are 5, 6, 7, and 8 bits per character. The ASCII character set requires 7 bits per character. The personal computer scan code mode requires 8 bits per character. Since most terminals require ASCII or personal computer scan code modes, the most often used parameters are 7 and 8 bits per character. Use 8 bits per character for personal computer scan code terminals.



All the terminals that MS OS/2 *MULTIUSER* supports require that the data bits be set to 8. However, because MS OS/2 *MULTIUSER* is capable of adding on terminals with new terminal drivers, the ability to change the number of data bits from 8 to 7, 6, or 5 is provided. When in doubt as to what number to place in the data bits field, select 8.

### **Parity:**

The parity mode provides data error checking at the data port. As data is received, the data port verifies the data according to the parity mode set. If the data is incorrect, an error condition is provided by the host system or terminal. When parity is enabled, an extra bit is added to the data character based on the parity mode (even, odd, mark, or space). This bit is added by the transmitter for the receiving port to validate the data. Disable parity for personal computer scan code terminals. Because MS OS/2 *MULTIUSER* provides support for only personal computer scan code terminals, the parity parameter should be set to none.

### **Stop Bits:**

The stop bit is used by the transmitter to indicate when transmission of a character is completed. The stop bit parameter specifies the bit spacing or delay at the end of each character prior to starting to transmit a new one. The stop bit choice has little effect for most host-to-terminal connections. This parameter will usually be 1.

### **Flow Control:**

Several handshaking options exist in the asynchronous interface. Handshaking can be divided into two types: software handshaking and hardware handshaking. Software handshaking techniques insert a special data character in the data stream to tell the other device to start or stop sending data. The handshaking character is usually sent when the

receiving device's buffer is full or the device is busy. This method is referred to as XON/XOFF or XPC. Use XON/XOFF for ASCII ports; use XPC with scan code terminals. Because MS OS/2 *MULTIUSER* supports only scan code terminals in the initial offering, this parameter should be set to XPC.

Another method of handshaking is via hardware control signals. Several control signals are provided within the (Electrical Industry Association, EIA) RS-232 interface. Typically, each device provides two output signals and from two to four input signals, depending on whether the device is Data Circuit Equipment (DCE) or Data Terminal Equipment (DTE). While each signal is defined by an EIA specification, the industry has provided variations of the signals used. While only the standard signals are described here, you will find that the CONFIG TERMINAL program will allow configuration of the host port for the necessary hardware handshaking for connecting the peripheral device. Following is a list of status and handshaking signals used by data ports:

<u>Signal Name</u>	<u>Input/Output</u>	<u>Signal Description</u>
TxD	output	Transmit Data
DTR	output	Data Terminal Ready
RTS	output	Request To Send
RxD	input	Receive Data
DSR	input	Data Set Ready
CTS	input	Clear To Send
RI	input	Ring Indicator
DCD	input	Data Carrier Detect

The output signals associated with receive data ready status are DTR and RTS. When a data port comes on line, the DTR signal is activated to establish that a data port is on line. When it is ready to receive data, the RTS signal is activated. If the data port deactivates the RTS signal due to a busy or a full data buffer, it is referred to as hardware handshaking.



Some data ports will not receive data unless these signals are active. When configuring host system data ports, the DTR and RTS signals should be set to the ON state if software handshaking is used. MS OS/2 *MULTIUSER* will activate the signals when it is ready for data. Terminals will activate these signals when they come on line.

The input signals associated with transmit data ready status are DSR and CTS. Before data is transmitted, these signals are examined for ready status. If hardware handshaking is enabled, the data port will not transmit data unless these signals are active.

The following are examples of data types and handshaking methods:

<u>Data Type</u>	<u>Handshaking Method</u>	<u>Application</u>
Receive personal computer scan code	XPC	User Terminal
Data transmit and receive	XON/XOFF	User Terminal
BINARY data transfer	Hardware	File Transfer

### Connection Type:

MS OS/2 *MULTIUSER* allows terminal configuration parameters to detect when a terminal comes on line or goes off line. Terminal on/off line conditions may occur when the terminal is switched off or its data cable is disconnected. As described earlier, certain signals become active when a device comes on line. The EIA RS-232 standard's signal for establishing a connection is DTR. While the industry follows this standard, the signals that become active at the opposite data port will depend on the cable wiring. The following is an example of how each signal should be connected in a host system-to-terminal configuration:

<u>Host System</u>	<u>Data Port</u>	<u>Terminal Data Port</u>
RxD	<-----	TxD
DTR	----->	DSR
RTS	----->	CTS
TxD	----->	RxD
DSR	<-----	DTR
CTS	<-----	RTS
RI	<-----	
DCD	<-----	

This example shows the optimum host system to terminal connection. There are variations due to cable types. In the example, the connection signal DTR for each data port is connected to the other data port's DSR. This allows either data port to detect an on/off line condition. If connect via DTR is selected in the CONFIG TERMINAL program, MS OS/2 *MULTIUSER* will be able to detect when a specific data port is in use. If a cable is used connecting transmit/receive data only, connect/disconnect and hardware handshaking options are not available.

The example shows RI and DCD as unconnected inputs. In a host system-to-terminal connection, these signals are not required; however, some cables are wired so that DTR connects to RI and DCD. For MS OS/2 *MULTIUSER*, this has no effect other than providing two additional terminal connect/disconnect options. When using cables that deviate from the standard DTE to DTE (host system to terminal) connection, care should be taken with CONFIG TERMINAL parameters.

If hardware handshaking is used, the signals used for handshaking should not be the same as the signals used for connect/disconnect. This would result in a signal to MS OS/2 *MULTIUSER* that the data port is disconnected when actually the terminal's buffers are full or not ready. Recommended cabling connection and alternate examples can be found in Appendix E, "Cable Types and Configuration Implications."

## Receive Buffer Size

The receive buffer size defines the size of the receive buffer allocated by the serial port device driver. This value defaults to 1024 and should never need to change.

## Configuring a Dial-In Terminal

This section describes the steps required to install dial-in ports on a MS OS/2 *MULTIUSER* system. It discusses the modem hardware requirements and the software configuration requirements.

A dial-in port is defined as a serial port on an MS OS/2 *MULTIUSER* system associated with a modem connected to a telephone line that can be used to remotely access the system.

A modem is used to convert the digital signals from a computer or terminal into signals that can be sent over telephone lines and convert them back into digital signals at the other end. Modems also handle the details of dialing and answering the telephone. In this section, the discussion will be limited to outside users calling into the MS OS/2 *MULTIUSER* host computer.

System Administrators should be aware of the security implications of allowing remote dial-in to the system. Make sure all logins are secured with *passwords* to prevent unauthorized access.

## Modem Hardware Requirements

The first step is to select the serial port to be used on the MS OS/2 *MULTIUSER* system. This port can be any serial port installed on your machine.

Once the port is selected, the next step is to obtain a modem and modem cable to attach to the port. The basic modem requirements are:

1. The modem must be configurable so that DCD and DSR signals are controlled by the line. These signals are raised when a call comes in and the connection is successfully established. When the call is terminated (either normally or due to a line error), these signals are dropped.
2. The modem should be switch-programmable rather than software-programmable, so the modem will remember its configuration across power failures. Software-programmable modems can be supported by manually programming them and saving the parameters into the modem's non-volatile memory. Consult the modem manual for details.



3. If the modem is software-programmable, it should use the standard Hayes command set, which is the modem standard. Software-programmable modems that do not use the standard Hayes command set can be used but will require user customization of modem control strings.

Once the modem and async port are set up, use a standard personal computer modem cable to connect the two together. For further details, refer to Appendix F, "Setting Up Specific Modem Types."

## Deleting a Terminal Profile

To delete a terminal profile to make a particular terminal inactive, follow the steps below:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line.
2. Move the selection bar over the *terminalname* that is to be deleted. The terminals are displayed beneath the associated terminal subsystem.
3. Press **F10** to go to the action bar, select Profile, then select Delete.
4. Press **ENTER** to delete the terminal.

## Renaming a Terminal Profile

Follow the steps below to rename a terminal profile:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line.
2. Move the selection bar over the *terminalname* that is to be renamed.

3. Press **F10** to go to the action bar, select Profile, then select Rename. The following panel appears:

```

Rename Terminal

Old terminalname ..... :SERIAL1

New terminalname ..... [           ]

Esc=Cancel    F1=Help

```

4. Enter the new *terminalname*.

## Moving a Terminal Profile

To move a terminal profile:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line.
2. Move the selection bar over the *terminalname* that is to be moved.
3. Press **F10** to go to the action bar, select Profile, then select Move. The following panel appears with a list of terminal subsystems that the terminal can be moved to:

```

Move Terminal to another Subsystem

Choose the terminal subsystem. Then press Enter.

Console

Esc=Cancel    F1=Help

```

4. Move the selection bar to the destination terminal subsystem and press **ENTER**.

## Copying a Terminal Profile

To copy a terminal profile:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line.
2. Move the selection bar over the *terminalname* that is to be copied.
3. Press **F10** to go to the action bar, select Profile, then select Copy. The following panel appears:

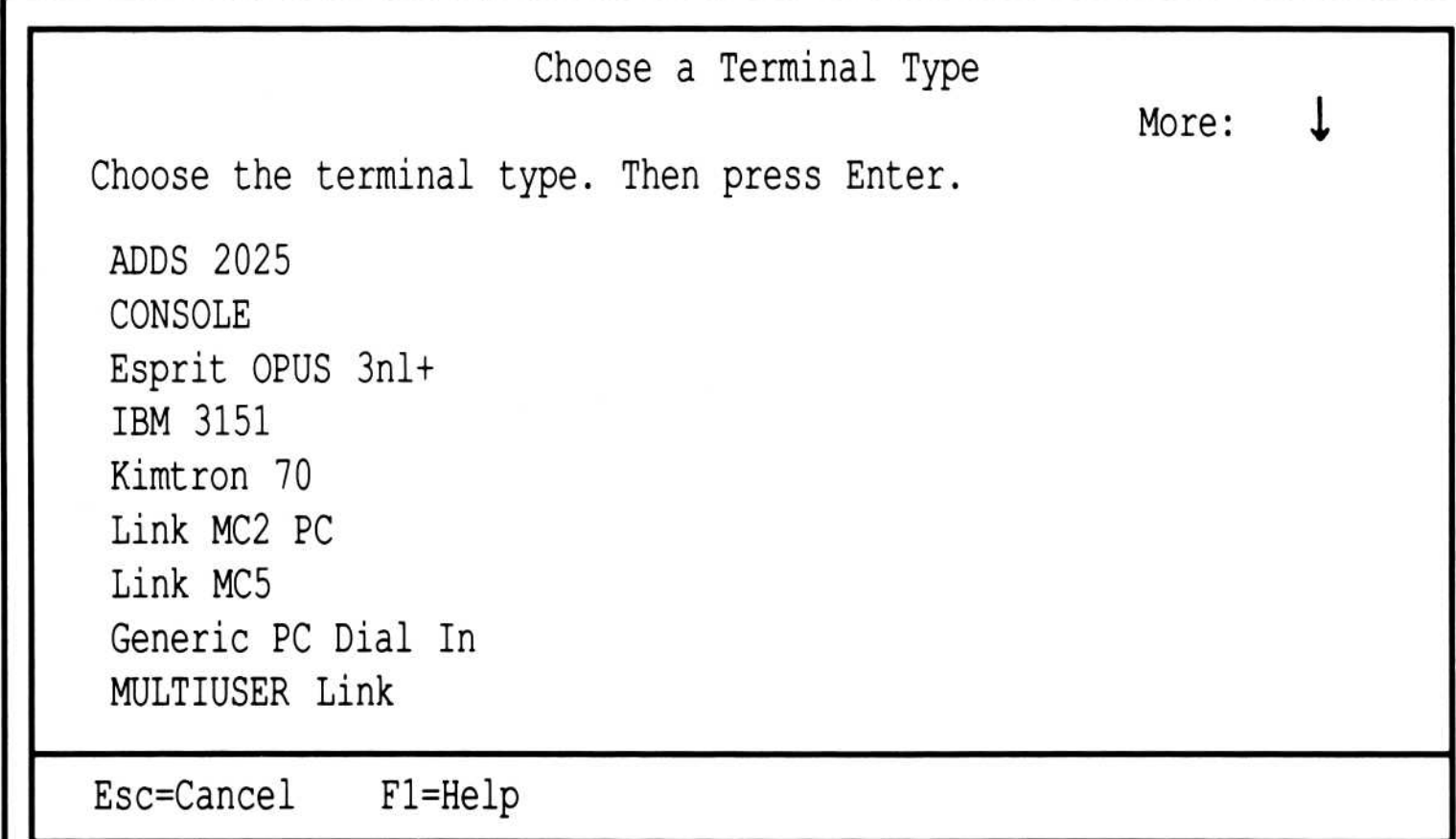
Copy a Terminal	
Original terminalname .....	:SERIAL1
New terminalname .....	[ ]
Esc=Cancel      F1=Help	

4. Type the new *terminalname* and press **ENTER**.
5. Move the selection bar to the device name that is to be selected and press **ENTER**.

## Changing a Terminal Type

To change a terminal type, follow the steps below:

1. Display the terminal configuration panel by entering the command **CONFIG TERMINAL** at the command line.
2. Move the selection bar over the *terminalname* that is to be changed and press **ENTER**.
3. Press **F10** to go to the action bar, select **Options**, then select **Change terminal type**. The following panel appears:



Choose a Terminal Type

More: ↓

Choose the terminal type. Then press Enter.

ADDS 2025  
CONSOLE  
Esprit OPUS 3nl+  
IBM 3151  
Kimtron 70  
Link MC2 PC  
Link MC5  
Generic PC Dial In  
MULTIUSER Link

Esc=Cancel    F1=Help

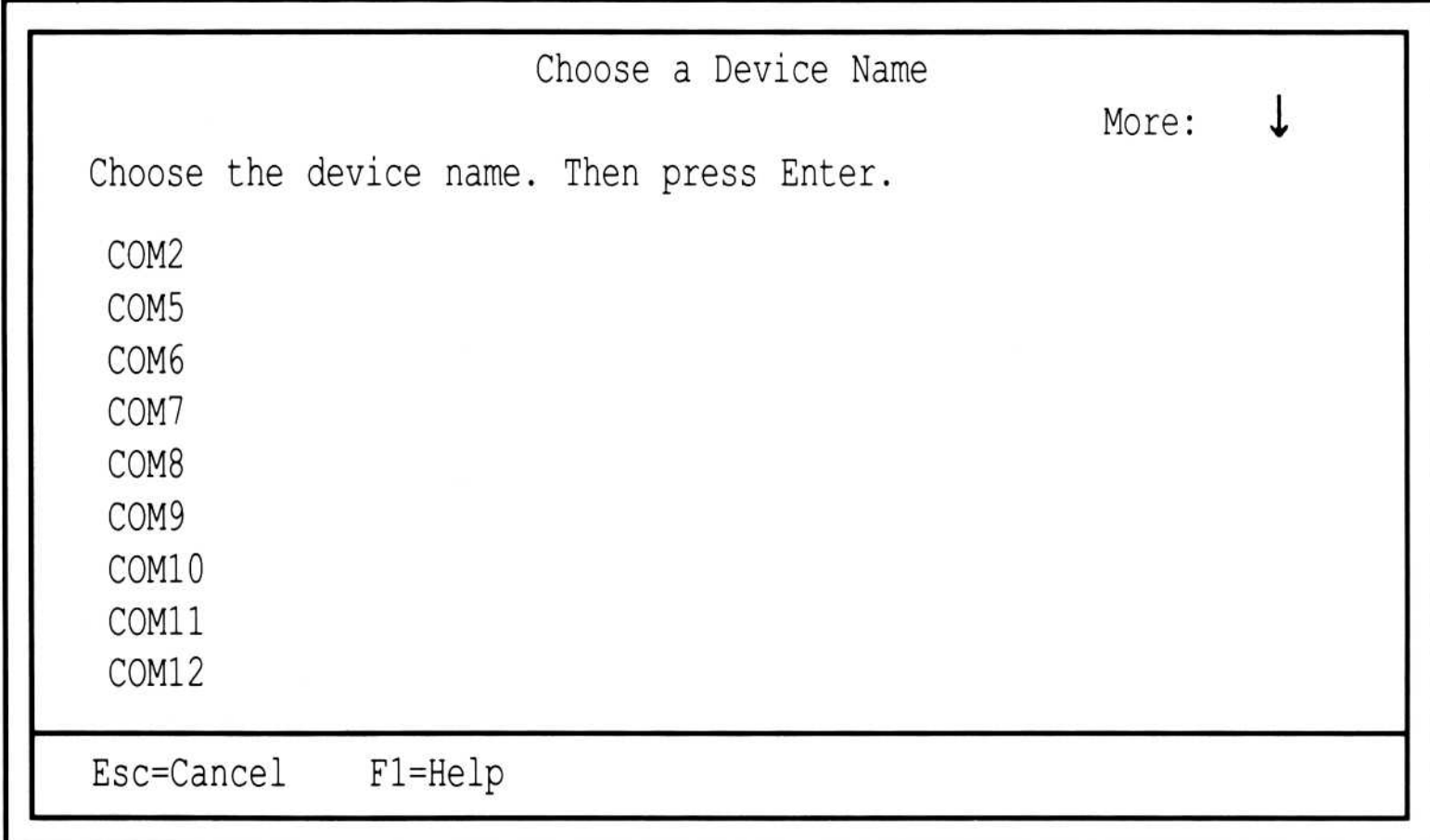
4. Move the selection bar to the terminal type that is to be selected and press **ENTER**.



## Changing Device Name

To change a device name:

1. Display the terminal configuration panel by entering the command CONFIG TERMINAL at the command line.
2. Move the selection bar over the *terminalname* whose device name is to be changed and press **ENTER**.
3. Press **F10** to go to the action bar, select Options, then select Change device name. The following panel appears:



Choose a Device Name

More: ↓

Choose the device name. Then press Enter.

COM2  
COM5  
COM6  
COM7  
COM8  
COM9  
COM10  
COM11  
COM12

Esc=Cancel    F1=Help

4. Move the selection bar to the device name that is to be selected and press **ENTER**.

## Using System Defaults

To change the autologin parameters and hotkey settings to the defaults for the system.

1. Display the terminal configuration panel by entering the command CONFIG TERMINAL at the command line.
2. Move the selection bar over the *terminalname* that is to be changed and press ENTER.
3. Press F10 to go to the action bar, select Options, then select Use system defaults.

## Using Terminal Type Defaults

To change all configuration parameters (with the exception being the autologin parameters and the hotkey settings) to the defaults for the configured terminal type:

1. Display the terminal configuration panel by entering the command CONFIG TERMINAL at the command line.
2. Move the selection bar over the *terminalname* that is to be changed and press ENTER.
3. Press F10 to go to the action bar, select Options, then select Use terminal type defaults.

## Resetting Terminal Configuration Parameters

To reset the configuration parameters to the settings when the last save was issued.

1. Display the terminal configuration panel by entering the command CONFIG TERMINAL at the command line.
2. Move the selection bar over the *terminalname* whose parameters are to be reset and press ENTER.
3. Press F10 to go to the action bar, select Options, then select Reset.

## Refreshing Configuration Parameters

Follow the steps below to refresh all the parameters to the most recently saved values:

1. Display the terminal configuration panel by entering the command `CONFIG TERMINAL` at the command line.
2. Press **F10** to go to the action bar, select Options, then select Refresh and press **ENTER**.

## Creating a New Terminal Subsystem

The following procedure explains how to create a new terminal subsystem. Refer to the documentation that came with the terminal subsystem adapter for information that will need to be entered in the "Create a New Terminal Subsystem" window.

To create a new terminal subsystem:

1. Display the Terminal Configuration menu by entering the command `CONFIG TERMINAL` at the command line.
2. Move the selection bar over the "System Profile" choice (topmost).
3. Press **F10** to go to the action bar, select Profile, then select New.

The following panel is displayed:

Create a New Terminal Subsystem	
Title .....	[ ]
Filename .....	[ ]
DLL .....	[ ]
First device name .....	[ ]
Number of devices .....	[ ]
Esc=Cancel    F1=Help	

5. Enter the new subsystem information (refer to the documentation that comes with the terminal subsystem adapter) and press **ENTER** when complete.

The new terminal subsystem is created. See "Adding a Terminal Subsystem" to make use of the new terminal subsystem that has been created.

## Reserving a Serial Port for Custom Use

Many communications programs require the use of a serial port at the device driver layer. To reserve a serial port for custom use, do not configure that port as a port with a terminal connected to it. If a port is not configured for use with a terminal, MS OS/2 *MULTIUSER* will not attempt to initialize the port. When the communications program attempts to open the port, the request succeeds allowing the custom use of the serial port.



# CHAPTER 7

## CONFIGURING SECURITY

Security provides a multitude of functions meant to ensure the user a private and reliable working environment within the MS OS/2 *MULTIUSER* system. First, through use of a *loginname*, access to the system is limited to only those allowed. Then, by controlling and adjusting file system *permissions*, each user is given a personal view of the system with private workspace. Also, through use of security *classes* each user is provided with a set of limitations to ensure that the integrity and privacy of the entire system remains intact.

MS OS/2 *MULTIUSER* can give programs their own identity so that they can take on different security characteristics from their users. This type of program is called a *registered program*.

MS OS/2 *MULTIUSER* provides additional security measures by protecting operating system functions from errant or malicious programs; thus the user's service cannot be interrupted by another user's login.

Security configuration is controlled primarily through the use of the CONFIG ACCESS utility. Some security parameters are associated with the user and group configurations; these are controlled using CONFIG PROFILES.

This section describes the fundamental concepts of security and provides descriptions of the utilities that apply to security. In addition, it describes the audit trail capabilities and the installation considerations of security. It begins with a brief description of the user, group and login concepts as they apply to securing your system.

## USERS AND GROUPS

A user on an MS OS/2 *MULTIUSER* system is a person who has formal permission to access the system facilities. This formal

permission is granted by creating a user profile using the CONFIG utility. The user profile maintains the following types of information:

- Personal data, such as name and telephone number.
- History, such as when the user was last on the system.
- Security data, such as *password* and class.
- Capabilities such as when the user can access the system and how much resource is available.

When a user profile is created, it is done under a formal and unique system name. This name is called a *username*.

The user profile also maintains group membership data. A group encompasses users with similar needs and characteristics. For example, a work group made up of 10 users may all be accessing the same applications and data, all with the same system requirements and security restrictions. To simplify the management of the profiles describing these 10 users, a *group profile* is created. The group profile is created and given a name (called a *groupname*) using the CONFIG utility.

The group profile basically contains default user data that pertains to all members of the particular group. For example, if the memory limit for the group is set at 1 megabyte (MB), each member's default restriction is 1 MB. These defaults can be overridden on a case-by-case basis by changing an individual's user profile.

All users must be a member of at least one group. If the installation has no real need for establishing groups, a single default group can be used for all user profiles and the users need not be aware of this default group.

## LOGIN

All MS OS/2 *MULTIUSER* system users must login to access any of the system facilities. The process of login accomplishes the following:

- Establishes the validity of the user by checking existence of user and group profiles and by verifying *passwords*
- Establishes a login session for the user based on the parameters given in the user and group profiles

In order to login, a user must provide a *loginname*. This *loginname* is a combination of the *username* and the *groupname*, input as *username.groupname*. Given the *loginname*, the system can establish any group defaults and then apply specific requirements for the individual.

As mentioned above, a user profile contains group membership information. A user can be a member of more than one group and can have different system characteristics, depending on the specific *loginname* used. However, in many cases a user may be a member of only one group, or a user may login under a specific group most of the time. For these cases, a default login group can be stated when configuring the user. This allows a user to enter only his or her *username* at the login prompt.

## Passwords

In addition to the *loginname*, it is typical to require a *password*. The System Administrator can define *password* requirements (for example, age, pattern, required/optional) using the CONFIG SYSTEM command. A user is prompted for a *password* if it is required for the user's *loginname*. After *password* entry, the system login process validates the



*loginname* and *password* combination. There are a number of reasons why a login is rejected: examples are bad or expired *password*, invalid or expired *loginname*, invalid login time, or the user is already logged in the maximum number of allowable times. The CONFIG USER and CONFIG SYSTEM functions control the login parameters and restrictions.

A user can control the user's *password* within the boundaries that the System Administrator has established through CONFIG SYSTEM. Passwords can be changed using the PASSWORD utility or by entering change information at the Login: prompt. If a user forgets the password, the System Administrator can change the *password* using CONFIG USER. The *password* cannot be queried.

## Autologin

It is possible to configure a terminal so that no actual login is required, thus allowing immediate access to an application. This is accomplished through the autologin feature of terminal configuration (see Chapter 6, "Autologin Parameters"). There is still a user and group (a *loginname*) and the system still goes through the login process; however, it happens automatically. No *passwords* are entered, resulting in the bypass of a fundamental security check.

## SECURITY CLASSES

MS OS/2 *MULTIUSER* provides a number of functions, primarily presented in the form of utilities, that can be categorized by their effect on the users of the system. Some functions, like changing the system date, affect all login sessions. Others can be used to change the capabilities of any user. There are functions that impact system global resources, such as formatting a disk partition. In order to control access to these "cross-user" functions, each *loginname* is assigned a security



classification, or class. The *Citrix MULTIUSER Command Reference* has the detailed security class restrictions for all the utilities.

The class is assigned and changed by the CONFIG USER function and can be one of the following:

- Administrator
- Operator
- User
- Guest

Each of these classes is described below.

## Administrator

The Administrator class is provided to allow individuals to perform system management and maintenance functions. Typical duties of an Administrator are:

- Create and update profiles.
- Install applications.
- Apply security measures.
- Review security and error logs.
- Perform system backup.
- Change hardware configurations.
- Manage system resource usage.

All of the utilities and file directories associated with these duties are available to a user who is configured with the Administrator security class.

As a general rule, an Administrator login has complete access to the entire system. However, for system integrity purposes, some Administrator duties should be performed while in maintenance mode, a mode in which no security functions are in place and only a single login is allowed. For more information on maintenance mode, see Chapter 13, "System Maintenance."

It is recommended that in a standard system installation users do not login under an Administrator *loginname* on a regular basis, but login only as an Administrator to perform one of these special functions. This prevents inadvertent modification of critical system data and provides a more secure and stable system for the entire user community.

## Operator

Operator security class is useful for an individual whose responsibility is to monitor system usage and help the users with problem determination and correction. Typical duties of an operator are:

- Review the error log for trends.
- Review the security audit for attempted security breaches.
- Make temporary resource changes to allow a user to complete a task.
- Manage print spooler queues.

The Operator class is usually restricted to making changes only to the runtime environment of the system, such as a temporary change to a user's memory limits. As a rule, these changes aren't permanent. However, as in the case of the print spooler queues, some permanent changes are allowed. Also, the Operator class has read access to all the facilities of MS OS/2 *MULTIUSER* to view the current status of the entire system.

It is not necessary that a facility have someone who is logged in as an Operator, but this is a convenient way to have read access to virtually the entire system without danger of making permanent changes.

## **User**

The User security class is the most common and is designed to give each user a private work environment. While logged in as "user", only utility functions that have no impact on other users can be executed. For example, a user can see resource data about himself or herself but cannot see other individual's resource utilizations. Because increasing user limits could potentially impact others, a User cannot change any limits.

The User class cannot modify configurations. The User class is restricted by the system from making changes that compromise system security or integrity.

The *Citrix MULTIUSER User's Guide* provides a comprehensive description of the functional capabilities of the User class.

## **Guest**

Guest class is quite limited in function but it is useful in the following situations:

- The *loginname* is provided for training.
- The *loginname* is provided for Guests (or temporary users).
- The *loginname* is part of a turn-key application and no changes should be required.

The Guest class is a subset of the User class and has no authority to make any configuration or runtime changes. Harmless query operations are allowed.

## SECURITY ATTRIBUTES

Named resources in MS OS/2 *MULTIUSER* can have certain security attributes. These attributes are similar to file system extended attributes and are used to describe various security characteristics of the resource.

Only HPFS partitions support security attributes. Because MS OS/2 *MULTIUSER* is always installed on an HPFS partition, security attributes can be maintained for almost all named resources. However, these security attributes cannot be maintained for files and directories that reside on a file system other than HPFS.

There are three types of security attributes: Access Control Lists, Audit Attributes, and Ownership. These are described below.

### Access Control List (ACL)

One security attribute is called an Access Control List (ACL). The purpose of an ACL is to maintain a list of users and their associated access permissions for the resource to which the



ACL attribute is attached. By providing a list, the Administrator can selectively control access to a resource down to the individual *loginname*.

An entry in an ACL contains two pieces of information: the identification of the login and the permissions associated with the login. The permissions indicate what types of operations (like read or write) are permitted and the identification relates to a specific *loginname*, a group of logins, a class name, or possibly an *aliasname*. (*Aliasname* is described later as a way to give a system identity to a program.)

Several ACL concepts simplify the definition and maintenance of security. These are:

- Implicit file system ACLs

An MS OS/2 *MULTIUSER* file system is a hierarchy of directories, subdirectories, and files within subdirectories. MS OS/2 *MULTIUSER* provides a file system security where all files and directories that do not have ACLs are governed by the parent access controls. These files and directories are said to have an implicit ACL.

- Global access names

Within an ACL there is a list of *loginnames*. This allows for granting access to a file or directory to a select set of system users. To simplify maintenance, the *loginname* can contain a global specification for the *username* and the *groupname*. For example, an entire group of users can be given equal access permissions to a resource by entering " \*.groupname " as the *loginname*.

- Full screen ACL maintenance

The CONFIG ACCESS utility is the mechanism for controlling the system resource access. This utility

provides a full screen interface to visually describe who has what type of access and allows a user to make the necessary changes without intimate knowledge of the ACL concept.

## Audit Attributes

Audit attributes indicate how a resource is to be audited. This essentially states what type of actions against the resource need to be logged for subsequent review in the security audit trail.

Unlike ACLs, the audit attribute for directories does not default to a parent; that is, no directory has an implicit audit attribute. However, for convenience, a file without a security audit attribute will take on the security audit attribute of the file's immediate parent subdirectory.

The Security Auditing section describes how the resource audit attribute interacts with other audit requirements to establish auditing at any given point.

Like the ACL, the audit attribute is added and changed using the CONFIG ACCESS utility.

## Ownership

Ownership applies only to HPFS files and directories and specifies the *username* of the current owner of the file or directory. Ownership is established at file or directory creation and can be changed at a later time using the OWNER utility. If the owner of a file or directory is deleted, the file or directory becomes unowned.

## SYSTEM RESOURCE SECURITY

### File System Security

A High-Performance File System (HPFS) partition is a global system resource, available to all users of an MS OS/2 *MULTIUSER* installation. In order to provide security, MS OS/2 *MULTIUSER* implements Access Control Lists for HPFS files and directories. The Access Control List (ACL) is a security attribute associated with a file or directory and contains user permission data.

Only HPFS partitions support ACLs, so other file systems such as the MS DOS-compatible FAT file system cannot be secured to this degree. A FAT file system partition can be secured at the drive letter level through device security. This is described below.

File system security is the key to isolating user data. When a user is logged in under a particular *loginname*, the user can only "see" those portions of the file system for which that *loginname* has access permissions. The user is not allowed to list (with a DIR) or change to (with a CD) a directory without proper permission.

Permissions relevant to the file system are described below, along with a description of the operations supported under each permission.

### Classes and File System Access

Classes are usually associated with user function points but a class name can be associated with other named resources such as files. This is primarily done at system installation to establish different views of the file system for different classes. For example, the Administrator class is given almost complete access to the entire file system. However, it is not

always true that an Administrator has complete access; it depends on how the ACLs are defined for the resources.

## Device Security and Serialization

System devices are secured in much the same way as the file system directories and files described above by using CONFIG ACCESS to place ACLs on devices.

System devices are:

- Fixed disk drive letters (such as C).
- Removable disk drive letters (like A).
- Network drive letters.
- Printers (for example LPT1).
- Other character devices.

When a user accesses a device, the ACL for that device is checked to ensure accessibility. Permissions for devices are described in the Permissions Section.

Some devices need to be serialized; that is, only one user at a time should be able to get to the device. For these devices, the Administrator should set up the access controls such that the users have Reserve (V) permission. This allows a user to access the device, but only after a RESERVE request has been issued by the user. Once the reserve request is granted, that user is the only one who can access the device until a RESERVE /D is issued to release the reservation. For example, use of a diskette drive should be serialized with the RESERVE utility to prevent any accidental conflicts in updating a diskette.



All permissions for a device that needs to be serialized should be set to V; otherwise, the serialization will not work properly. If you have U permission, you need not reserve the device; therefore, you can bypass the serialization mechanism.

As a default at install time, drives A and B are set such that all users must RESERVE them before use.

## Printer Security

A typical MS OS/2 *MULTIUSER* installation manages the printers using the print spooler. This allows multiple users to concurrently print to a printer port (say LPT1) without concern for intermixing the reports on the physical printer. Because of this, a spooled printer should not be set up to be serialized using the V permission. However, use of the printer can be controlled by updating the ACL for the printer, giving Use (U) permission to those authorized to access the printer.

The capabilities for query and update of the spooler queues are associated with the security class of the user. See the SPOOL utility in the *Citrix MS OS/2 MULTIUSER Command Reference*.

## Terminal Security

Terminal security provides a means of controlling access to terminals and dial-in ports. With terminal security, an Administrator can define who can have login access to a configured terminal. Again, this security is managed by using CONFIG ACCESS to associate an ACL with a given terminal name. The permissions associated with terminal security are described later.

There are two forms of security for a terminal. One is a Login (L) permission which can be put in the ACL associated with a terminalname. Another is the ability to limit the class of the user which can use a terminal, thus preventing Administrator level functions from some terminals if desired. This is done using CONFIG TERMINAL.

## User Functions

Class control of the various system functions is controlled by management of "user function points". These are named objects describing the function (the name can describe an object or a verb). The user function points have Access Control Lists (ACLs) that can be modified, but this is rarely necessary.

User Function Points (UFPs) are of two types, verbs and objects. A verb is an action function; that is, a UFP name indicates the specified function. For example, a verb-type UFP might be "delete\_a\_user." For these types of UFP, only Execute (X) permission makes sense.

A UFP as an object has four permissions: Query-self (Q), Query-other (O), Modify-self (M), and Modify-other (T). The name of the UFP is the object of the action indicated by one of these permissions.

## Program Interfaces and Functions

Application Program Interfaces (APIs) and other program functions can be secured through the same ACL mechanisms. These topics are addressed in the subsequent sections.

## REGISTERED PROGRAMS

MS OS/2 *MULTIUSER* implements a program interface level of security to:

- Secure the MS OS/2 *MULTIUSER* programming interface.
- Allow a program to have an access permission that is different from the permissions of the user of the program.

These special capabilities are given to select programs through a process called program registration. Using the REGISTER utility, an Administrator can give an identity to a program or set of programs. This identity is called a registered program aliasname, or just aliasname. This *aliasname* is then used in defining access controls (ACLs) for programs that need special security rights.

REGISTER and CONFIG ACCESS describe the mechanisms for establishing the program security. Further detail on these capabilities is provided below.

### Securing the MS OS/2 *MULTIUSER* Programming Interface

The MS OS/2 *MULTIUSER* Application Programming Interface (API) provides a wide range of functions any program can perform. A number of these functions can potentially disrupt the operational integrity of the entire system or allow a program to bypass standard security measures. In order to prevent a malicious or errant program from improperly using these functions, they are "secured." Securing these functions involves associating an ACL with the function and placing the authorized registered program *aliasnames* in the ACL.

One example of a secured function is the ability to have direct access to a disk partition, bypassing the file system. If a



program has this capability, it can inadvertently destroy all user data or it can look at all data on the disk, whether the data is secure or not.

During installation, a standard set of registered programs is established to allow the MS OS/2 *MULTIUSER* system programs to function in this secured environment. No other program access the potentially damaging API functions. However, there may be some instances where it is necessary to give an application access to certain secured system functions. In this case, the Administrator can register the program using the REGISTER utility and give its *aliasname* access permission to the function using CONFIG ACCESS.

## Securing MS OS/2 *MULTIUSER* Program Functions

A number of programming functions lie underneath the MS OS/2 *MULTIUSER* API. In some cases, it is not appropriate to secure the API but it is necessary to secure the underlying function. These functions are called Program Function Points (PFPs); they are established during system installation.

It may be necessary for you to allow a program to access one of these secured functions. In this case, you need to REGISTER the program and update the ACL for the appropriate PFP (using CONFIG ACCESS) so that the *aliasname* you registered can Execute (X) the PFP.

## Access Permissions for Programs

Some applications require access permissions different from the permissions of the user of the application. For example, an application may require access to data or control files that its users shouldn't have access to. In cases such as this, an



Administrator can register the program and place its *aliasname* in the ACLs for the files or directories to which the program must have access. Then, regardless of the *loginname* of the user, this program can always have access to the data.

## SECURITY AND PROGRAM LOADING

Several security barriers are established to prevent an unauthorized execution of a program (*.EXE*), a dynalink library (*.DLL*), or access to a protected programming interface. Much of these access checks are made during program installation.

The first check that is made deals with locating the *.EXE* being requested using the PATH environment variable. If the request to run a program does not include the path of the *.EXE*, PATH is used to search for the *.EXE*. The basic security on the file system will prevent the loader search sequence from accessing paths that the user cannot access. The result is that the loader skips non-accessible paths.

Once the *.EXE* is actually found, the MS OS/2 *MULTIUSER* loading process can cause indirect access to a dynalink library, which in turn can cause another indirect access to a different *.DLL*. Security for program execution requires that the "caller" of the program have Execute (X) access. Security also requires that the caller have Execute access to the individual application programming interfaces (APIs) being used in the dynalink, as well as the MS OS/2 *MULTIUSER* system interfaces (the DOSCALLS) being used.

The net result is that even though a user has access to an *.EXE*, the *.EXE* still may not run because it is not set up to access all the other *.DLLs* and APIs that it needs.

## Security Checks Made During the Load Process

During the load process for each *.EXE* or *.DLL* being accessed, the following checks are made:

- The program or dynalink is checked for Execute (X) access to each *.DLL* it needs.
- It is checked for Execute (X) access to each programming interface it uses within the *.DLL* it is accessing.
- It is checked for Execute (X) access to each system interface it requires.

**NOTE:** Although a program may already be loaded and in use by one user, it still goes through the load process and security access is still checked if it is invoked by another user.

Note also that while this seems like a rather extensive process, usually there are very few secured APIs and the access checks are a natural extension of the dynamic interface linking process and file system loading process already existing in the MS OS/2 *MULTIUSER* loader.

## What Happens When a Load Fails

What happens if a security access fails depends on the circumstances under which it failed.

1. If the *.EXE* is in a directory you cannot Read (R) or Execute (X) and a user (or the executed program) have not specified an explicit path for the *.EXE*, a "not found" error occurs. When running *CMD.EXE*, the message will indicate that the command is not valid.

2. If a user does not have Execute (X) access to the *.EXE* that the user is attempting to invoke from the *CMD.EXE* command line, an "access denied" error is displayed.
3. If a user is running another program (not *CMD.EXE*) and that program executes a system request to load another program, an error ("access denied," error code 5) will be returned to the application attempting the load request. The user's response will depend on how the application handles the error.
4. If a user has X access to an *.EXE* but the *.EXE* does not have X access to a *.DLL* it loads (or if the *.DLL* it loads does not have X access to all its *.DLLs*, and on and on), an "access denied" error will be returned (or displayed).
5. If the *.EXE* or *.DLL* is denied access to an API, the system will generate a popup indicating that the *.DLL* it is trying to access is in error. The error generated in the popup is SYS0005 (access denied).

## How to Correct a Loading Problem

You must first determine the source of the problem. Try to identify what program is causing the failure and what type of failure it is. If it is not obvious, you can review the audit trail for XD (Execute Denied) accesses. If you don't normally enable this audit event, you can use CHANGE EVENTS and CHANGE AUDIT to temporarily turn them on and re-execute the load attempt. If it is a case where the *.EXE* is just not found, review the PATH environment variable and the LIBPATH in your *CONFIG.SYS* file.

Once you have established the source of the problem, you can correct it using the following guidelines.



1. If it is a "not found" problem, use Execute access to the necessary path (using CONFIG ACCESS) or move the program(s) to a path to which the user has access.
2. If the problem is access denied during loading of an *.EXE* from *CMD.EXE*, use Execute access to that *.EXE* or to a parent path.
3. If an access denied error is detected because an application is attempting to load an *.EXE*, or an *.EXE* or *.DLL* is attempting to load another *.DLL*, the application being run may not be set up right. To correct this, study the relationships of the *.EXEs* and *.DLLs*, then REGISTER the *.EXEs* or *.DLLs* which need special X access to a path or file, and finally update the ACL on the path or file (CONFIG ACCESS) to allow the aliasname you register to have X access.
4. If access is denied to an API (resulting in a popup) and if it is valid that the dynalink or program have access to that API, REGISTER the program or dynalink and update the ACL associated with the API (using CONFIG ACCESS) to include X access for the aliasname you registered.

## RESTRICTIONS ON CHANGING PERMISSIONS

The mechanism for changing permissions of resources is CONFIG ACCESS. This utility is available without limitation to the Administrator security class. However, there are often cases where a user needs the ability to change permissions of his or her files or directories in order to share data with other users. This is accommodated by allowing a user to update permissions on files and directories for which that user is the owner. Refer to the Security Attributes section for a description of owner.



A person logged in under the owner *username* can change the permissions of the file or directory. If the resource is not owned, an Administrator must change the permissions or change the owner.

## SECURITY PERMISSIONS

This section describes all of the security permissions associated with the different types of resources that can be secured in MS OS/2 *MULTIUSER*. Each permission is given a unique ID in the form of a single letter. This letter is shorthand in the utilities that query and set permissions. The following are two additional shorthands applying to all resource types:

<u>ID</u>	<u>Name</u>	<u>Description</u>
*	All	This ID gives the user all of the permissions that apply to the particular type of resource being targeted. This is only a shorthand method used by the utilities and is not maintained internally.
N	Null	This indicates that the user should get NO or NULL permissions. This equates to the absence of all permissions and is a method of excluding access.

### File System Permissions

File system permissions apply to files and directories. These permissions have slightly different meanings depending on what the target resource is, a file or a directory. Remember, even if the target resource is a file with no ACL, the file's permission is still governed by its implicit ACL that comes from a parent directory entry.

<u>ID</u>	<u>Name</u>	<u>Description</u>
R	Read	Provides the ability to read and copy a file, change to a directory, or view the contents of a directory. This does not grant Execute (X) permission to a file.
W	Write	Provides the ability to update a file. This does not grant permission to truncate a file, since file truncation is considered to be a create operation.
C	Create	Provides the ability to create a file or subdirectory. Once a file directory is created, it can be written to until the file is closed, at which time Write (W) permission is required for further updates.
D	Delete	Provides the ability to delete a file or remove a subdirectory. Delete permission on a directory entry does not grant permission to delete that entry.
X	Execute	Provides the ability to execute the program. This does not allow you to read or copy the program.
A	Attributes	Provides the ability to change the file or directory attributes.

## Device Permissions

<u>ID</u>	<u>Name</u>	<u>Description</u>
U	Use	Provides the ability to use the device.
V	Reserve	Provides the ability to reserve the device for exclusive use.

## Terminal Permissions

<u>ID</u>	<u>Name</u>	<u>Description</u>
L	Login	Provides the ability to login using the target terminal.

## API Permissions

<u>ID</u>	<u>Name</u>	<u>Description</u>
X	Execute	Provides the ability for a registered program to execute the API.

## Program Function Point Permissions

<u>ID</u>	<u>Name</u>	<u>Description</u>
X	Execute	Provides the ability for a registered program to execute the function point.

## User Function Point Permissions

<u>ID</u>	<u>Name</u>	<u>Description</u>
S	Query Self	Provides the user with the ability to query the object specified by the UFP name, but only that which is associated with the user's loginname.
O	Query Other	Provides the ability for a user to query the object specified by the UFP name. A user can query all other users' UFP objects.

<u>ID</u>	<u>Name</u>	<u>Description</u>
M	Modify Self	Provides the ability for a user to modify the object specified by the UFP name. A user can modify only the user's objects.
T	Modify Other	Provides the ability for a user to modify the object specified by the UFP name. A user can modify any user's object.
X	Execute	Provides the ability for the user to execute the action described by the UFP name.

## DISABLING SECURITY ACCESS CHECKS

While security facilities such as CONFIG ACCESS and auditing are always enabled, enforcement of the security permissions can be disabled. You may need to do this to perform maintenance tasks such as system data backups or to recover from a situation where you have locked yourself out of the ability to execute CONFIG ACCESS. You may also wish to do this because you do not wish to enforce security at your installation.

The two ways to disable access checks are:

1. Enter maintenance mode

In this case, only one user can be logged in and there are no access checks.

2. Set SECURITY=OFF in *CONFIG.SYS*



Setting security off disables all access checks regardless of what ACLs are present and how many users are logged in. Every user has complete permission to see and do anything on the system.

**WARNING:** The system integrity is jeopardized in SECURITY=OFF, allowing users to access, modify, and delete profiles, issue any system commands, and execute any programs. This environment is usually only recommended for the "closed" installations; that is, the ones that run only one application.

Disabling access checks does not disable any login or *password* requirements.

## SECURITY AUDITING

MS OS/2 *MULTIUSER* is capable of tracking security events. The targets of the tracking and the events to be tracked are established by the System Administrator.

### Audit Events

An audit event is defined as a combination of user intent to access a facility and the system's allowance of the user's access intention. For example, if a user intends to change a file and the system denies this change, the event is "write-denied."

The part of the audit event corresponding to access intent equates directly to the permissions that can be granted to a resource. In the example above, a user might have RWCD permissions to access a file. However, the "event" in the example is the act of writing the file; thus, the intent portion of the event is W.

The other portion of the event is the allowance; that is, whether the intent is granted or denied. In the example, because the user had RWCD permissions, the access would succeed. Therefore the entire event is described as WG, or write-granted.

Specific audit events are described as a combination of the access intent (or permission requested) and the result, granted (G) or denied (D).

## Audit Targets

The following are potential targets of a security audit:

- Entire system - The CONFIG SYSTEM utility is used to specify full system audit requirements.
- A *username* - The CONFIG USER utility is used to audit users.
- A *groupname* - The CONFIG GROUP utility is used to audit groups.
- A file or directory - The CONFIG ACCESS utility is used to audit files and directories.
- A device or a terminal - The CONFIG ACCESS utility is used to audit devices and terminals.
- Any other named resource that can have an ACL  
The CONFIG ACCESS utility is used to audit other resources such as APIs and function points.

For any given event, more than one security audit may apply; in this case, auditing is additive. As an example, say a user is being audited for all granted and denied attempts to delete a

file (DG and DD). If the system audit specifies all denied attempts to write a file (WD), the resulting audit will be addition of all audits in affect (DG+DD+WD).

## Audit Log

The security audit data is maintained in a system log as part of MS OS/2 *MULTIUSER* event management. All maintenance of the data and report generation is done using the EVENTS utility. Regardless of any audit attributes placed throughout the system, event logging must be enabled. By default, during installation, event logging for audit events is enabled. Refer to Chapter 14, "System Event Logging" for details.

## Secondary Audit Events

There are additional security related events that are not covered under the access audit described above. One example of this is the attempt to login with an invalid password. These are logged under a separate system event (S).

## SECURITY ATTRIBUTES AND INSTALLATION

During system installation of a new MS OS/2 *MULTIUSER* system, you can indicate how you want to set up the initial security environment. This includes:

- Initial login for the System Administrator and a few users. At least one Administrator login is required so that the user and terminal configurations can be completed. Install also provides a simplified way to create a few additional users.

- **System audit attribute.**  
The global system audit attribute can be specified during install or it can be postponed and entered in the ways described above.
- **Security Access Check Enabling.**  
Security checks can be disabled. This selection can be made during installation. The default is security enabled.

## Resource Security Layout

The standard security configuration establishes a rigid environment in which a User class login has only fundamental access to the file system and the system utilities. The layout is always established during system install but can be changed or enhanced at a later time by using CONFIG ACCESS. As mentioned previously, the ACL checking can be disabled; however, the install process still establishes this standard layout.

The layout of the install drive file system is established with the following permissions:

	<u>Administrator</u>	<u>Operator</u>	<u>User</u>	<u>Guest</u>
\	*	R	N	N
\USR	*	R	N	N
\OS2	*	RX	X	X
\OS2\CTX	RX	RX	N	N
\OS2\DLL	*	RX	X	X
\OS2\DLL\CTX	RX	RX	N	N
\OS2\SYSTEM	*	RX	N	N
\SYSCTX	R	R	N	N
\SYSCTX\DEV\				
default	*	*	*	*
\SYSCTX\TERM\				
default	*	*	*	*
\USR\xxx	*	R	see note below	



**NOTE:** Several users can be established during install. The user directories (\USR\xxx) are configured so that the associated user has complete access. For the example directory \USR\DAVIDH, the loginname DAVIDH.\* would have "\*" access.

## **What Happens to the \USR\xxx Directory**

When a user is created, a home directory is placed in the \USR\xxx directory. This user directory has an ACL that is created by taking the ACL from the parent (\USR) directory and adding the user.\* entry to the ACL. This gives the associated user complete access to all permissions within the user's working directory.

A new user is not given any additional permissions other than those currently existing for all users in the group.

## **PRESERVING SECURITY ATTRIBUTES**

Security attributes for files and directories are attached to the file and directory entries. The following describes various file operations that may affect the state of the security attributes:

- **COPY and XCOPY**

When a file is copied to another location, the security attributes of the file are not copied. If the file is placed in a secured HPFS directory, access to the new file will be governed by its implicit ACL.

- **RENAME**

Rename of a file or directory preserves an explicit ACL and preserves the ownership.

- **MOVE**

Move is a general case of rename.

If the MOVE is within the same drive letter, ownership is preserved. If the file or directory has an explicit ACL, this ACL will be preserved.

However, if the file or directory has no explicit ACL, the move may cause its implicit ACL to change by causing the file to be moved to a different parent subdirectory.

#### ■ BACKUP and RESTORE

These utilities preserve the explicit ACLs and the ownership. This is the standard method of backing up a directory path. If only a portion of a directory tree is backed up and this portion of the tree does not have explicit ACLs, no ACLs are saved on diskette. During the restore process, the directory and file will take on the security attributes of the closest parent with an explicit ACL.

#### ■ System Editor

System Editor preserves an explicit ACL on a file and the ownership of the file.

#### ■ Other Editors

Some editors may lose a file ACL by virtue of the way they operate. Many will create a new file under a temporary name, delete the old file, and rename the new to the old name. A similar problem exists with the extended attributes of a file. Users should take care when editing files with ACLs using an editor other than System Editor.

## USING THE CONFIG ACCESS UTILITY

The CONFIG ACCESS utility can be used to configure file system, device, and terminal security attributes. To start the utility type in CONFIG ACCESS and the following menu appears:

Configure File and Directory Access	
Options	Select
Access Name: ADMINISTRATOR	
Permissions	
{UV}	- C:
[RWCDXA]	└ - \
[RWCDXA]	└ + OS2
[RWCDXA]	└ + SYSCTX
{RWCDXA}	└ - USR
[RWCDXA]	└ + RICHA
[RWCDXA]	└ + ANDYS
[RWCDXA]	└ + KURTP
[RWCDXA]	└ + CHRISL
[RWCDXA]	└ + JOELS
\$ [RWCDXA]	└ + JEFFK
[RWCDXA]	└ + MIKEH
[RWCDXA]	└ + EDJ
[RWCDXA]	└ + ANNM
[RWCDXA]	└ + MIKED
[RWCDXA]	└ + BRADP
[RWCDXA]	└ + GREGG
[RWCDXA]	└ + SCOTTK
F1=Help   F3=Exit   F5=Refresh   F6=Change Permissions   F10=Actions	

**NOTE:** This utility is consistent with the other Full Screen Utility Interfaces (FSI). If you are not familiar with the FSI interaction, you should review Chapter 1 in the Command Reference, "Navigating Through the Full Screen Utilities."

Illustrated is a representation of the file system for drive C and the security attributes associated with each tree member (drive C, root, and its subdirectories) for the Administrator class.

## Access Name

The Access Name is shown under the action bar and can be one of three things:

1. It can be a *loginname* (for example, RICH.A.CITRIX).
2. It can be a Registered Program Aliasname (for example, MY\_PROGRAMS).
3. It can be a Class Level: Administrator, Operator, User, or Guest.

## Expanded and Compressed Directories

A "+" in front of a tree member indicates that it is compressed (i.e. its subdirectories are NOT shown) and that it can be expanded. A "-" in front of a tree member indicates that it is expanded (its subdirectories ARE shown) and that it can be compressed. A straight line in front of a tree member indicates it does not have any subdirectories and cannot be expanded or compressed. Any directory can be expanded to see its files by expanding the directory to its fullest and pressing ENTER.

To expand a tree member, cursor to the member and press ENTER.

To compress a tree member, cursor to the member and press ESC.



## **Permissions**

Contained within the brackets are the permissions for the Access Name, in this case Administrator.

## **Explicit Versus Implicit ACL**

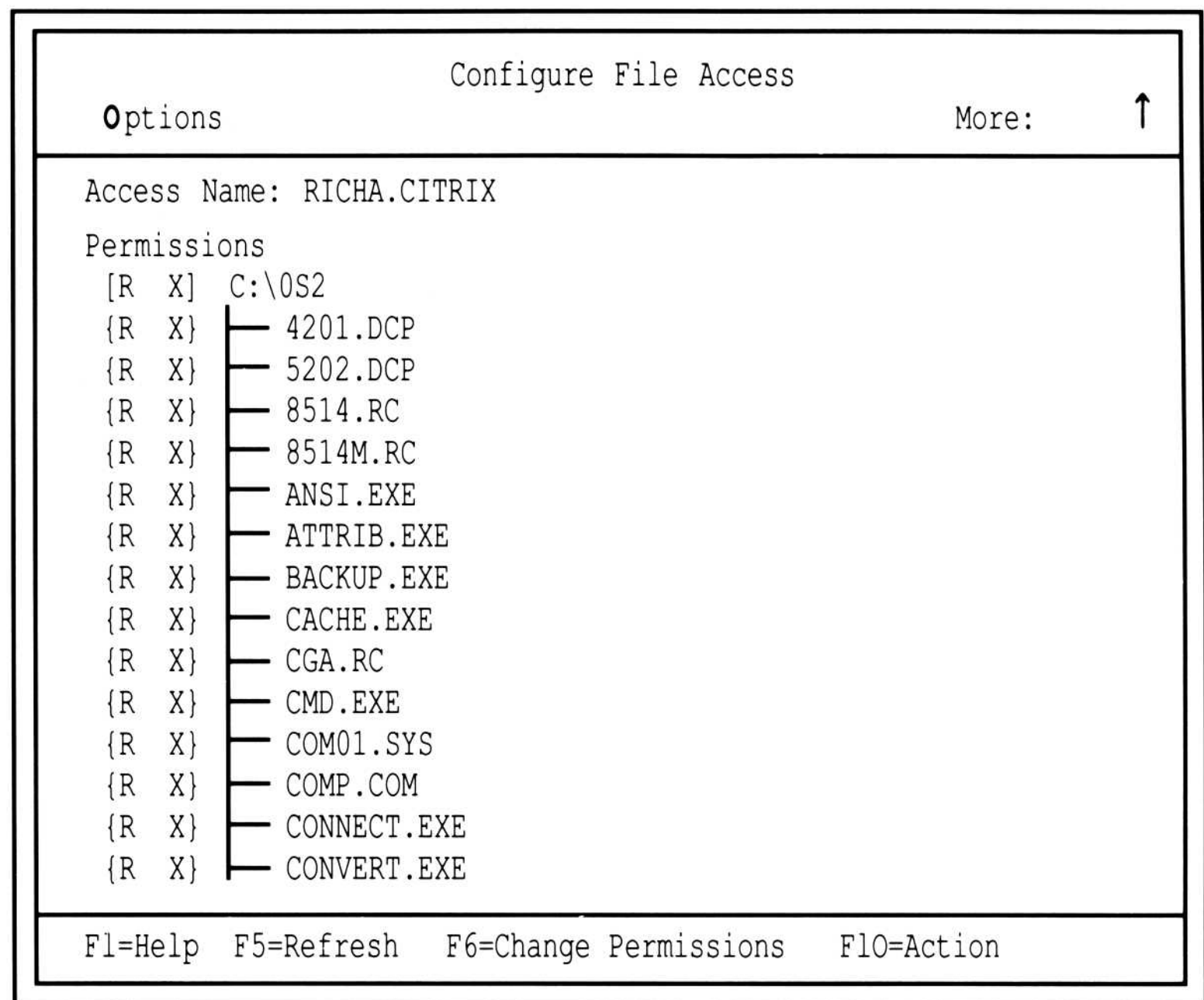
Notice the different kinds of brackets in front of the tree members. The "[...]" brackets indicate that an explicit ACL governs this resource. The "{...}" brackets indicate that an implicit ACL governs this resource (that is, the ACL is inherited from a parent directory).

## **Audit Security Attribute Indicator**

Notice the "\$" in front of the \USR\JEFFK permission brackets. This indicates that this resource has a specific audit security attribute associated with it.

## **Viewing Files Within a Directory**

To view the files and its permissions, move the cursor to the directory tree member and press **ENTER**. The directory will expand either to subdirectories or files. Continue to expand the directory entry using **ENTER** until the files are displayed. The files and its permissions for the given Access Name are shown in the same format. See below.



From the options pulldown you can "View or Change an ACL," "View or Change Audit," or "Remove Audit." These functions are identical to the options pulldown for the "Configure File and Directory Access" menu and will be described there.

## Refreshing the Menu (F5)

The directory and files displayed is a replica of the system at the time the utility is invoked. To refresh the menu, press F5. The menu is redisplayed.

## Viewing the Configuration Options

Pulldown the "Options" action bar item from the "Configure File and Directory Access" menu.

Configure File and Directory Access	
Options	Select
Change Access Name . . . .	
View or Change ACL . . . .	
View or Change Audit . . . .	
Remove Audit	
Change Drive . . . .	
Expand All	
Change Path . . . . .	
[RWCDXA]	+ RICH
[RWCDXA]	+ ANDYS
\$ [RWCDXA]	+ KURTP
[RWCDXA]	+ CHRISL
[RWCDXA]	+ JOELS
[RWCDXA]	+ JEFFK
[RWCDXA]	+ MIKEH
[RWCDXA]	+ EDJ
[RWCDXA]	+ ANNM
[RWCDXA]	+ MIKED
[RWCDXA]	+ BRADP
[RWCDXA]	+ GREGG
[RWCDXA]	+ SCOTTK

F1=Help   F3=Exit   F5=Refresh   F6=Change Permissions   F10=Actions

## Changing the Access Name

Select "Change Access Name" if you want to see the permissions of other users, classes, or register program aliases for the directory tree. A sample display panel follows:

```

                                Configure File and Directory Access
Options   Select

Access    Change the Access Name
Perm      [RWC]
{UV}      Access Name ..... [RICHA.CITRIX ]
[RWC]     ESC=Cancel   F1=Help   F6=List Names
[RWC]
{RWC}
[RWC]
[RWC]
+ CHRISL
[RWCDXA] + JOELS
[RWCDXA] + JEFFK
$ [RWCDXA] + MIKEH
[RWCDXA] + EDJ
[RWCDXA] + ANNM
[RWCDXA] + MIKED
[RWCDXA] + BRADP
[RWCDXA] + GREGG
[RWCDXA] + SCOTTK

F1=Help   F3=Exit   F5=Refresh   F6=Change Permissions   F10=Actions

```

Type in the Access Name in the space provided and press **ENTER**. The name selected replaces Access Name on the "Configure File and Directory Access" menu.

## Changing the Permissions for Multiple Directories or Files

You can change the permissions for an accessname for many directories or files. To do this select the accessname you wish to change the permissions for (see "Changing the Access Name"). Select the directories of files you wish to change permissions for by moving the cursor to the entries and pressing the **SPACEBAR**. A ">" will appear next to each entry selected. Press **F6** while in the "Configure Directory Access" or "Configure File Access" menus. The following popup will appear:



Configure File and Directory Access		
Options	Select	
Access Name: *.CITRIX		
Permissions	<div style="border: 1px solid black; padding: 5px; float: right; width: 30%;">           Change Permissions  <input checked="" type="checkbox"/> Read  <input type="checkbox"/> Write  <input type="checkbox"/> Create  <input type="checkbox"/> Delete  <input type="checkbox"/> Execute  <input type="checkbox"/> Attribute  <input type="checkbox"/> None  <input type="checkbox"/> All  <hr/>           F1=Help    Esc=Cancel         </div>	
{UV}	C:	
[N    ]	└─ \	
[N    ]	└─ + OS2	
[N    ]	└─ + SYSCTX	
{N    }	└─ - USR	
> [N    ]	└─ + RICHA	
> [N    ]	└─ + ANDYS	
> [N    ]	└─ + KURTP	
> [N    ]	└─ + CHRISL	
> [N    ]	└─ + JOELS	
> \$[N    ]	└─ + JEFFK	
> [N    ]	└─ + MIKEH	
> [N    ]	└─ + EDJ	
> [N    ]	└─ + ANNM	
> [N    ]	└─ + MIKED	
> [N    ]	└─ + BRADP	
> [N    ]	└─ + GREGG	
> [N    ]	└─ + SCOTTK	
F1=Help    F3=Exit    F5=Refresh    F6=Change Permissions    F10=Actions		

Choose the desired permissions and press **ENTER**. The permissions will be changed.

## Viewing an ACL

Select "View or Change ACL" if you want to see ADD, VIEW, or change the ACL associated with the resource. A sample menu follows:

Options

View or Change an ACL

Resource Name: C:\OS2

Type	Permissions	Access Name
Class	[RWCDXA ]	Administrator
Login	[R X ]	*.*

F1=Help F10=Actions

The Access Control list is shown, along with the full path of the resource name. The ACL consist of three columns:

1. Entry type (that is, *loginname*, class level, or register program alias)
2. Permissions (R=Read, W=Write, ..., and so forth)
3. Access Name

Note that the same notation is used to indicate if an ACL is explicit ("[...]") or implicit ("{...}").

From the "Options" pulldown of this menu, you can add or change an ACL, remove an ACL entry, or delete the ACL altogether.

To remove an ACL entry, select "Remove ACL Entry" from the "Options" pulldown. The entry is removed from the list (see restrictions below).

To delete an ACL altogether, select "Delete ACL" from the "Options" pulldown. The ACL will be replaced with its parent ACL (see restrictions below).

**RESTRICTIONS:** The "Remove ACL Entry" and "Delete ACL" functions can only be executed on an explicit ACL; that is, an ACL that has "[...]" versus "{...}".

To change an ACL entry, cursor to the entry you want to change and then select "Change an ACL Entry" from the "Options" pulldown. The following menu will be displayed:

Add or Change an ACL Entry	
Resource .....	:C:\OS2
Access Name .....	[Administrator ]
Permissions .....	(x) Read (x) Write (x) Create (x) Execute (x) Delete (x) Attributes ( ) None ( ) All
F1=Help F6=List F10=Actions	

The entry selected will be displayed.

To change the entry, select the permissions desired and press **ENTER**.

To add a new entry, select "Add an ACL Entry," type in the Access Name desired, select the desired permissions, and press **ENTER**.

**NOTE:** Adding or changing an ACL Entry to an implicit ACL (that is, an ACL with "{...}") will force the system to create an explicit ACL (that is, an ACL with "[...]") using the implicit ACL as a base.

## Viewing the Audit Security Attribute (ASA)

Select "View or Change AUDIT" if you want to see ADD, VIEW, or change the audit security attribute associated with the resource. The following menu is displayed:

View or Change the Audit Selections			
Resource : C:\OS2			
Audit flags:			
Read .....	( )	Grants	( ) Denials
Write .....	(x)	Grants	(x) Denials
Create .....	( )	Grants	(x) Denials
Execute .....	( )	Grants	(x) Denials
Delete .....	( )	Grants	( ) Denials
Attribute .....	( )	Grants	( ) Denials
F1=Help			



To add or change the audit security attribute (ASA), select the desired grants and denials and press **ENTER**. A "\$" should appear next to the permission letters, that is, "\$[...]" on the "Configure File and Directory Access" menu indicating the resource has an ASA.

**NOTE:** A tree member does not have to show a "\$" to get to the "View or change Audit Menu." You can place an audit security attribute on a resource via this menu.

**NOTE:** A "\$" will be placed by the tree member when you add an audit security attribute. The "\$" will be removed when the audit security attribute is removed.

## **Removing The Audit Security Attribute**

Select "Remove Audit" if you want to remove the Audit Security Attribute (ASA). The "\$" next to the permission letters (that is, "\$[...]" on the "Configure File and Directory Access" menu) should be cleared. This indicates that ASA has been removed.

## **Configuring Other Drives**

Select "Change Drive" if you want to configure access for another drive. A menu will appear asking you to enter the drive that is to be configured. Enter a letter (A-Z) and press **ENTER**. The directory structure for the drive will be displayed.

## **Expanding All Subdirectories**

Select "Expand All" if you want to expand the selected directory to its fullest.

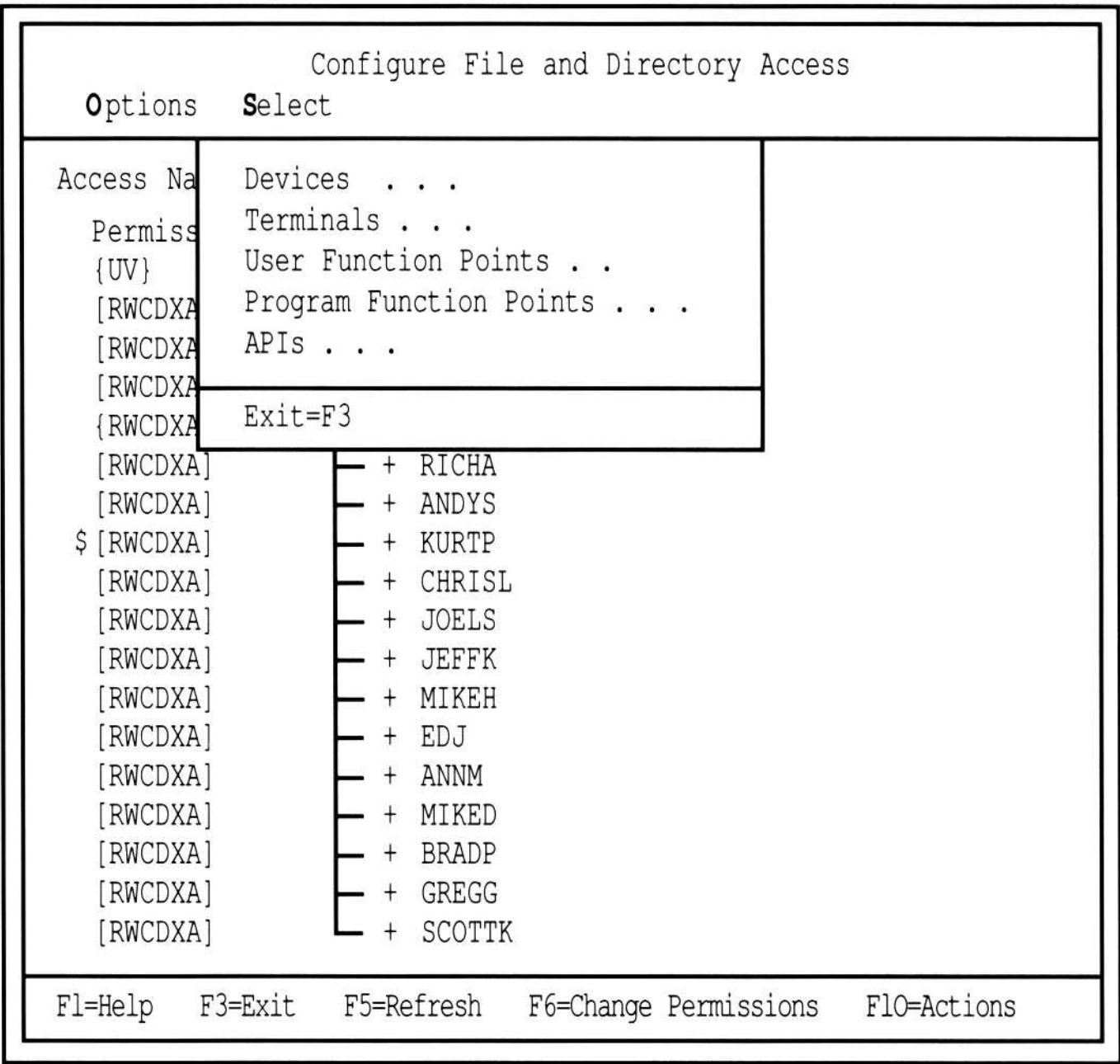
**NOTE:** This function is useful for users who want to view directories that normally can't be seen because the path to the directory can't be seen.

Changing the Path

Select "Change Path" if you want to change the selected directory you are currently working in.

Configuring Security Attributes for Resources Other Than Files and Directories

Pulldown the "Select" action bar item from the "Configure File and Directory Access" menu. Choose the select pulldown if you want to configure access to Devices, User Function Points "\UFP\," Program Function Points "\PFP\," Application Programming Interface "\API\," or Terminals "\TERM\".



All these menus are similar to the "Configure File and Directory Access" menu and contain the same functions where they apply. They are not shown to avoid redundancy.



# CHAPTER 8

## CONFIGURING PRINTING AND SPOOLING



## INTRODUCTION

The SPOOL utility is used to configure the printing environment for the MS OS/2 *MULTIUSER* system. When MS OS/2 *MULTIUSER* is installed, the spooler is automatically enabled, and the spooler process is automatically started when the system is booted.

If the spooler has been disabled, starting the SPOOL utility automatically starts the spooler process.

The SPOOL utility has both a command line interface and a full screen interface.

## PRINT QUEUES

Type SPOOL and press **ENTER** at the command prompt to start the full screen SPOOL utility. Using the full screen interface, you can perform the following actions on a print queue:

### Adding a Print Queue

Use the following steps to add a print queue:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Queues. A panel displaying the current print queues appears.
4. Press **F10** to go to the action bar.
5. Select Options.
6. Select Add Queue.

The following panel is displayed:

Add Print Queue			
Name .....	[ ]		
Printer Port .....	(x) LPT1		
	( ) LPT2		
	( ) LPT3		
Other Port(s) .....	[ ]		
Job Ordering .....	(x) FIFO		
	( ) Job Size		
Separator File ..	[DEFAULT.SEP		]
Print After .....	[12:00]	(X) AM ( ) PM	
Print Until .....	[12:00]	(X) AM ( ) PM	
Queue Processor.	[LMPRINT.DLL		]
Comment .....	[ ]		
ESC=Exit F1=Help			

The Add Print Queue panel needs the following information:

**Name:**

The name of the print queue may be from 1 to 12 characters long.

**Printer Port:**

This is the printer port associated with the print queue. You may choose more than one printer port.

**Other Port(s):**

If you have other printer ports besides LPT1, LPT2, and LPT3, type the port name or names here. If there are more than one, separate them with a comma. For example: COM10, COM14.

### Job Ordering:

This determines the sequencing order in which print jobs are printed.

1. Select FIFO to print jobs in "first in - first out" order; that is, in the order in which the print jobs were received by the spooler.
2. Select Job Size to print jobs in order of ascending print job size. Smaller jobs are printed before larger ones.

### Separator File:

This is the name of the separator file for the queue. The full path name must be listed. The separator file defines a separator page that will be printed before each print job.

Type DEFAULT.SEP to use the system default separator file.

Type NONE to have no separator page printed.

Refer to the following section on separator pages for more information.

### Print After:

### Print Until:

Determines the time that a print queue will be active for printing. The print queue will only print starting from the "print after" time until the "print until" time. If a queue receives a print job during a time when it is not active for printing, it will hold the jobs in the queue until it becomes active for printing again.

**Queue Processor:**

The queue processor removes print jobs from the print queue and manages their printing. The default queue processor is displayed. To use another queue processor, type in the full path name of the desired queue processor.

**Comment:**

The comment is for descriptive purposes only and does not affect the operation of the print queue. The comment may be up to 48 characters long.

## Changing a Print Queue

Use the following steps to change a print queue:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Queues. A panel displaying the current print queues appears.
4. Highlight the print queue you wish to change.
5. Press **F10** to go to the action bar.
6. Select Options
7. Select Change Queue

A panel similar to the one shown above is displayed. The only field you cannot change is the print queue name.

## Deleting a Print Queue

Use the following steps to delete a print queue:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Queues. A panel displaying the current print queues appears.
4. Highlight the print queue you wish to delete.
5. Press **F10** to go to the action bar.
6. Select Options.
7. Select Delete Queue.

## CONFIGURE A COMMUNICATION PORT

If you have a printer that is connected to a communication (serial) port, you may wish to change the port's mode. See the instructions that come with the printer to get the proper mode settings.

Use the following steps to configure a communication port:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Communication Ports. The Communication Port panel illustrated below appears.



Communication Port Details			
Device .....	[                      ]		
Baud Rate .....	( ) 300	( ) 2400	( ) 9600
	( ) 600	( ) 3600	( ) 19200
	(x) 1200	( ) 4800	( ) 38400
	( ) 1800	( ) 7200	( ) Other
Word Length .....	( ) 5	( ) 7	(x) 8
Parity .....	( ) even	( ) odd	(x) none
Stop bits .....	(x) 1	( ) 1.5	( ) 2
Handshake .....	( ) Hardware	(x) None	
ESC=Exit F1=Help			

The Communication Port Details panel needs the following information:

**Port Name:**

The name of the port the printer is connected to.

**Baud Rate:**

This is the rate at which data is sent and received. If the baud rate is different from what is shown, select Other and you will be prompted for a baud rate.

**Word Length:**

This is the length (in bits) of a byte of data.

**Parity:**

Parity is used to check whether the byte that arrived is the same as the one that was sent.

**Stop Bits:**

Delimits the end of each byte of data. Normally, one stop bit is needed.

HandShake:

Communication between the devices:

#### HARDWARE

Indicates the communication is controlled by electronic signaling.

#### NONE

Indicates the exchange is controlled by the program's exchange of information.

## CONFIGURE A PRINTER PORT

If you have a printer that is connected to a parallel port, you may wish to change the port's mode. See the instructions that come with the printer to get the proper mode settings.

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Printer Ports. The Printer Port panel illustrated below appears.

Printer Port Details	
Printer Port .....	<input checked="" type="radio"/> LPT1 <input type="radio"/> LPT2 <input type="radio"/> LPT3 <input type="radio"/> Other
Characters Per Line .....	<input checked="" type="radio"/> 80 <input type="radio"/> 132
Lines Per Inch .....	<input checked="" type="radio"/> 6 <input type="radio"/> 8
ESC=Exit F1=Help	

The Printer Port Details panel needs the following information:

**Printer Port:**

Select the port to change. If you have other printer ports besides LPT1, LPT2, and LPT3, you may select the option: Other. You will be prompted to type in the name of the other port.

**Characters Per Line:**

You may select either 80 or 132 characters per line.

**Lines Per Inch:**

This controls the vertical spacing. You may select either 6 or 8 lines per inch.

## SPOOL OPTIONS

### **Enable/Disable Spooling**

Use the following steps to enable/disable spooling:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Spooler Options. The Spooler Options panel illustrated below appears.

Spooler Options	
Spooler .....	<input checked="" type="checkbox"/> Enabled <input type="checkbox"/> Disabled
Spool Path .....	[C:\SPOOL ]
ESC=Esc F1=Help	

The Spooler Options panel needs the following information:

**Spooler:**

The spooler can either be enabled or disabled.

Enabled: if the spooler is enabled, the system will start the spooler process when the system powers up.

Disabled: if the spooler is disabled, the system will not start the spooler process when the system powers up. However, the spooler may be started at anytime by invoking the SPOOL utility. When the SPOOL utility is started, it checks to see if the spooler process is running; if it is not, the spooler process starts running.

## Setting the Spool Path

The spool path is where the print queues are maintained on disk. Use the following steps to set the spool path:

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Spooler Options. See previous panel.

**Spool Path:**

To change the path, type in a full path name.

**Setting the Refresh Rate**

The refresh rate determines how often the SPOOL utility's panels are automatically updated with current system data. Use the following steps to set the refresh rate.

1. Press **F10** to go to the action bar.
2. Select Setup.
3. Select Refresh Rate. The following Refresh Rate panel appears:

Refresh Rate
Spooler data refresh: Refresh Rate ..... [5 ] seconds
ESC=Exit F1=Help

**Refresh Rate:**

Type in the new refresh rate. This must be between 0 and 65535 seconds. A refresh rate of 0 means no automatic refreshing will occur. This new refresh rate is only used until you Exit the SPOOL utility.





## CREATING A SEPARATOR PAGE FILE

You can create your own separator page file instead of using the default. Separator page files use escape codes that give instructions to the printer.

Escape codes always start with an escape character, which can be any character you select, and end with a letter or number. The first line of the separator page file must contain only the escape character.

A table of escape codes that can be used for the separator page file follows:

<u>Code</u>	<u>Function</u>
@N	Prints the <i>username</i> of the person who submitted the job.
@G	Prints the <i>groupname</i> of the person who submitted the job.
@J	Prints the job ID.
@Q	Prints the queue name.
@D	Prints the date of printing.
@T	Prints the time of printing.
@L <i>text</i>	Prints text.
@F <i>pathname</i>	Prints the contents of file specified by <i>pathname</i> .
@H <i>nn</i>	Sets a printer specific control sequence where <i>nn</i> is a hexadecimal number sent directly to the printer. <b>NOTE:</b> To find the numbers to use see the printer manual.
@W <i>nn</i>	Sets the width of the separator page (default is 80).
@ <i>n</i>	Skips <i>n</i> lines. The range is 0-9. Skipping 0 lines goes to the next line.
@O	Prints the host name.

<u>Code</u>	<u>Function</u>
@B@S	Begins printing in single width block characters. Text is printed this way until @U is encountered.
@B@M	Begins printing in double width block characters. Text is printed this way until @U is encountered.
@U	Turns off block character printing.
@E	Ejects a page from the printer.

The default MS OS/2 *MULTIUSER* separator page file, *DEFAULT.SEP*, follows:

```
@
@B@S
@N@0
@G@4
@Q
@4
@U
@LUserName: @N@0
@LGroupName: @G@0
@LQueueName: @Q@0
@LFile Spooled: @D @L @T @0
@LJob ID: @J@0
@E
```

This creates a separator page like the following:

```

@@      @@      @@      @@@@      @@@      @@      @@@@@@
@@@    @@@    @@@@    @@      @@      @@      @@      @@
@@@@@@@@    @@      @@      @@      @@      @@      @@
@@@@@@@@    @@      @@      @@@@@    @@@@@    @@@@@@
@@  @  @@  @@@@@@@@@    @@      @@      @@      @@      @@
@@      @@  @@      @@      @@      @@      @@      @@      @@
@@      @@  @@      @@      @@@@    @@      @@      @@@@@@

```

```

@@@@      @@@@@@      @@@      @@@      @@@@@
@@      @@      @@      @@      @@      @@      @@
@@@      @@      @@      @@      @@      @@      @@
  @@@      @@@@@@      @@      @@      @@      @@      @
@@      @@      @@      @@      @@      @@      @@
@@      @@      @@      @@@@    @@@@    @@@@@@

```

```

@@@@      @@      @@@@@      @@@@@@@@@      @@@@@@      @@@@@
  @@      @@@@@      @@      @      @@      @@      @@      @@
  @@      @@      @@      @@@@@      @@@@@@      @@@@@@      @@
  @@      @@      @@      @@@@@      @@@@@@      @@@@@@      @@
  @@      @  @@@@@@@@@      @@@@@      @@@@@      @@@@@@      @
  @@      @@      @@      @@      @      @      @@      @@@@@
@@@@@@@@      @@      @@      @@@@@      @@@@@      @@@@@      @

```

UserName: MARKB  
 GroupName: SPOOL  
 QueueName: LASERQ  
 File Spooled: 01-19-90 02:22pm  
 Job ID: 27





# CHAPTER 9

## USING SYSTEM CONFIGURATION (*CONFIG.SYS*) COMMANDS

## INTRODUCTION

When you start MS OS/2 *MULTIUSER*, your system reads configuration commands from the *CONFIG.SYS* file on your start-up disk. These commands give information to the operating system about the setup, or configuration, of your particular system. This section gives the form of the MS OS/2 *MULTIUSER* configuration commands and examples of how they might be used. Even though most System Administrators will never need to change their *CONFIG.SYS* file, changes can be made, if necessary, by referring to the information in this section. For more information about the individual configuration commands, see the *Citrix MULTIUSER Command Reference*.

During installation, MS OS/2 *MULTIUSER* creates the *CONFIG.SYS* file; you can view the default values assigned to some of the configuration commands and modify some of those values at this time. Once the installation is complete, you can change command values, either by running the installation program again or by using a text editor to edit your *CONFIG.SYS* file. If MS OS/2 *MULTIUSER* detects an error in the *CONFIG.SYS* file, it displays an error message and pauses until you press the **ENTER** key. If you do not want MS OS/2 *MULTIUSER* to pause after displaying a *CONFIG.SYS* error message, add the following command to your *CONFIG.SYS* file:

PAUSEONERROR=NO

When you add this command, MS OS/2 *MULTIUSER* still displays the error message but it does not pause before processing the *CONFIG.SYS* file.

**NOTE:** All examples of *filenames* and directory names in this section use the High-Performance File System (HPFS).

## DEFINING THE *CONFIG.SYS* FILE

In order to run MS OS/2 *MULTIUSER*, certain configuration commands are required in your *CONFIG.SYS* file. During installation, MS OS/2 *MULTIUSER* adds these required commands to your *CONFIG.SYS* file. The following list shows you what some commands in the *CONFIG.SYS* file might look like after MS OS/2 *MULTIUSER* is installed:

```
SET COMSPEC=C:\OS2\CMD.EXE
IFS=C:\OS2\HPFS.IFS
RUN=C:\OS2\CACHE.EXE /LAZY:ON
LIBPATH=C:\OS2\DLL;C:\OS2\DLL\CTX;C:\;
SET PATH=C:\OS2;C:\OS2\CTX;C:\OS2\SYSTEM;C:\;
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\OS2\INSTALL;C:\;
SET PROMPT=[$P]
HOSTNAME=SYSTEM01
BUFFERS=30
MAXWAIT=3
MEMMAN=SWAP,MOVE
PROTECTONLY=YES
SWAPPATH=C:\OS2\SYSTEM 512
THREADS=512
COUNTRY=001,C:\OS2\SYSTEM\COUNTRY.SYS
DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
REBOOT=OFF
RESOURCE=ON
MAINTENANCE=OFF
CODEPAGE=437,850
DEVICE=C:\OS2\COM01.SYS
SECURITY=ON
```

The following list tells you what the configuration commands in this example do:

**SET COMSPEC.** Specifies the full path name of the MS OS/2 *MULTIUSER* command interpreter.

**IFS.** Initializes an installable file system.

**RUN.** Runs a program during system initialization.

**LIBPATH.** Specifies the search path for dynamic-link libraries.

**SET PATH.** Specifies a search path used to find executable programs.

**SET DPATH.** Specifies a data search path; used by programs to locate associated data.

**SET PROMPT.** Determines the appearance of the command prompt.

**HOSTNAME.** Specifies the name of the system.

**BUFFERS.** Specifies the number of disk buffers in memory.

**MAXWAIT.** Specifies the maximum amount of time any active process must wait before it is run.

**MEMMAN.** Specifies whether memory swapping and moving is permitted.

**PROTECTONLY.** MS OS/2 *MULTIUSER* must run in protect mode.

**SWAPPATH.** Specifies the location of the disk-swap file and the minimum free space (in kilobytes).

**THREADS.** Specifies the maximum number of threads that the system supports.

**COUNTRY.** Defines country-dependent information, such as time, date, and currency conventions.

**DEVINFO.** Specifies the information a device needs in order to use a particular code page.

**REBOOT.** Specifies whether MS OS/2 *MULTIUSER* is to enable the system restart and system dump hotkeys at the console.

**RESOURCE.** Specifies whether resource management should enforce user limits.

**MAINTENANCE.** Specifies whether MS OS/2 *MULTIUSER* operates in maintenance mode or normal mode.

**CODEPAGE.** Specifies which code pages your system is prepared to use.

**DEVICE.** Sets up a device driver.

**SECURITY.** Specifies whether security access checking is enabled or disabled.

## MODIFYING VALUES OF CONFIGURATION COMMANDS

You can use the System Editor to modify the *CONFIG.SYS* file. Changes take effect when you restart your system. Before you make changes to the *CONFIG.SYS* file, you should first make a backup copy of the file. The backup copy of the *CONFIG.SYS* file is useful if you make a change that causes an error and prevents your system from starting correctly. If such an error occurs, follow these steps to start your system:

1. Insert the MS OS/2 *MULTIUSER* Install disk into drive A and turn on your computer. If your computer is already on, restart it using the *SHUTDOWN* command.



2. Insert Disk 1 when prompted and press **ENTER**.
3. When the first panel appears, press the **ESC** key. This takes you to the command prompt.
4. Copy your backup copy of the configuration file (that you made before you changed *CONFIG.SYS*) to the root directory of the start-up drive. The copy you put on the start-up drive must be named *CONFIG.SYS*.
5. Remove the MS OS/2 *MULTIUSER* Disk 1 from drive A and restart your computer by pressing **CTRL+ALT+DEL**.

It is important to note that usually you will never need to change the values in the *CONFIG.SYS* file and that it is recommended that you not do so. If you need to change the *CONFIG.SYS* file, however, the information provided in this section should help you to understand how the configuration commands work.

## ENABLING INPUT/OUTPUT PRIVILEGE LEVELS

You enable or disable input/output privilege levels by using the IOPL configuration command.

The privilege level assigned to a program determines which data segments it can access and which machine instructions it can execute. Applications are usually assigned privilege level 3, which allows them to refer only to their own data segments and restricts them from issuing input/output instructions.

Applications that are assigned privilege level 2 can issue input/output instructions. When you specify IOPL=YES, a program that has been assigned privilege level 2 is allowed to execute machine instructions necessary to access a

particular input/output device. You can also specify input/output privilege levels for programs on an individual basis. This is the recommended method.

For example, to grant a printer driver named *NEWPRINT* access to a particular piece of hardware, add the following command to your *CONFIG.SYS* file:

```
IOPL=NEWPRINT.DRV
```

It is not necessary for you to specify *IOPL=NO*, since *NO* is the default value. Avoid using *IOPL=YES* because the MS OS/2 *MULTIUSER* security features are rigidly restrained. *IOPL=YES* allows any application program with privilege level 2 to access input/output instructions. For example, if the application program *123.EXE* needs *IOPL*, the operating error message is "SYS0197: the operating system is not presently configured to run this application" is displayed. If it is determined that this program should be allowed *IOPL*, you should add it to the list of *IOPL* programs:

```
IOPL=NEWPRINT.DRV,123.EXE
```

## MODIFYING MULTITASKING FEATURES

MS OS/2 *MULTIUSER* provides several configuration commands that let you modify its multitasking features. Although usually you will not need to use these commands, there may be times when you are advised to change these features to enhance how applications run with MS OS/2 *MULTIUSER*. The following commands let you modify the multitasking features of MS OS/2 *MULTIUSER*:

■ THREADS

■ MAXWAIT

- PRIORITY
- TIMESLICE

These commands are described in the following sections.

## Specifying the Maximum Number of Threads

To specify the maximum number of threads you can create at one time, use the THREADS configuration command.

MS OS/2 *MULTIUSER* allocates at least one thread for every running program. A program may request additional threads if it needs them to perform tasks simultaneously.

The THREADS configuration command sets a system-wide limit on the number of threads that can be created. During installation, MS OS/2 *MULTIUSER* sets the number of threads to 512. If you will be running many programs that use a large number of threads, you may want to use the THREADS configuration command to increase the number of threads that can be created at one time. To use this command, type THREADS= followed by the maximum number of threads you want to create at one time. You can specify any value from 64 through 1024.

For example, to increase the number of threads from 512 (the default value) to 572, add the following line to your *CONFIG.SYS* file:

```
THREADS=572
```

Be aware that as the number of threads is increased, a small amount of memory is used up.

## Specifying the Maximum Time a Process Waits

In a multitasking environment, each process that runs is assigned a priority level that determines how often the process can run. If a process has a high priority, it is granted permission to run more often than a process with a low priority. Thus, a low-priority process might wait a long time before it is allowed to run. You can change the amount of time a process must wait by using the *MAXWAIT* configuration command.

The *MAXWAIT* command specifies the maximum amount of time any active process must wait before it runs. If the specified time elapses, the priority of the waiting process is increased for one execution cycle (also called a time slice). To use the *MAXWAIT* command, you type *MAXWAIT=* followed by the number of seconds a process must wait before receiving a priority increase. You can specify any value from 1 through 255 seconds. The default value is 3 seconds.

For example, suppose that you want to make background programs, which normally have low priority, run faster. You could specify that you want a process to wait only one second before receiving a priority increase, by placing the following line in your *CONFIG.SYS* file:

```
MAXWAIT=1
```

Note that this will increase the speed of low-priority processes (such as background programs) but that it will also decrease the speed of high-priority processes (such as foreground programs).

Unless you are specifically advised to do so, it should not be necessary to change this setting. This command interacts with the *PRIORITY* command described in the next section.



The **PRIORITY** command must be set to **DYNAMIC** for the **MAXWAIT** command to have an effect on system performance.

## Disabling Dynamic Priority Assignments

In MS OS/2 *MULTIUSER*, processes (threads) are divided into three categories according to their priority: general, time-critical, and low. The general-priority category is further divided into foreground, background, and interactive subcategories. MS OS/2 *MULTIUSER* automatically adjusts the priority levels of general-priority processes as circumstances change (for example, when a foreground process becomes a background process), thus causing the process to run more often or less often. This feature is known as dynamic priority scheduling.

By default, MS OS/2 *MULTIUSER* allows dynamic priority scheduling; however, you can turn this feature off (or turn it back on) by using the **PRIORITY** configuration command. To disable dynamic priority processing and allocate CPU time to the processes according to the priorities they are currently assigned, type the following:

```
PRIORITY=ABSOLUTE
```

By default, the **PRIORITY** command is set to **DYNAMIC**. Unless you are specifically advised to do so, it should not be necessary to change this setting. This command interacts with the **MAXWAIT** command described in the preceding section. The **PRIORITY** command must be set to **DYNAMIC** for the **MAXWAIT** command to have an effect on system performance.

## Setting the Time-Slice Values

To specify the minimum and maximum amount of time that MS OS/2 *MULTIUSER* will dedicate to a given process before it



checks on other processes that may be waiting, use the `TIMESLICE` configuration command. A time slice is an interval of time used by MS OS/2 *MULTIUSER* to schedule the threads of a process. To use this command, type `TIMESLICE=` followed by the minimum and, optionally, the maximum time-slice value (in milliseconds). If you specify only the minimum time-slice value, the maximum time slice is also set to that value.

Unless you are specifically advised to do so, it should not be necessary to change this setting.

## MANAGING SYSTEM MEMORY

The following MS OS/2 *MULTIUSER* configuration commands help your system manage memory efficiently:

- `DISKCACHE`
- `MEMMAN`
- `SWAPPATH`
- `BUFFERS`

These commands are described in the following sections.

### Setting Up a Disk Cache

You can enable disk caching and specify the number of kilobytes of memory allocated for the disk cache by using the `DISKCACHE` configuration command. Since reading from a disk cache in memory takes less time than reading from storage, using a disk cache makes your system-response time much faster. When disk caching is enabled and a program instructs the system to read from your fixed disk, the system stores what it reads in the disk cache in memory. When the

program instructs the system to read from the fixed disk again, the system checks the cache for the data and reads from the fixed drive only if the data is not in the cache. To allocate a 128K disk cache in your system, add the following line to your *CONFIG.SYS* file:

DISKCACHE=128

If there is no DISKCACHE command in your *CONFIG.SYS* file, MS OS/2 *MULTIUSER* does not enable disk caching and no cache space is allocated.

This command is specific to partitions that are formatted using the FAT-based file system. If your system does not have any FAT-based partitions, this command should not be used.

## Enabling Memory Swapping and Moving

You can use the MEMMAN configuration command to enable or disable the memory-management techniques of swapping and moving.

Swapping is a technique by which some segments in memory are written to a file on a disk, making the memory these segments were using available for another purpose. The swapped segment is reloaded into memory when it is needed.

Moving is a technique by which MS OS/2 *MULTIUSER* can combine unused blocks of storage into larger areas, which can then be used by a program or application.

To force MS OS/2 *MULTIUSER* to bring all of the segments of all of the programs (that are executed) into physical memory and keep them there, add the following line to your *CONFIG.SYS* file.

MEMMAN=NOSWAP

This technique will ensure optimal system responsiveness. However, there must be enough physical memory to hold all the executing programs in the system.

You can use the MEMMAN command in several forms, including the following:

MEMMAN=NOSWAP,NOMOVE

MEMMAN=SWAP

MEMMAN=SWAP,MOVE

MEMMAN=NOSWAP

MEMMAN=NOSWAP,MOVE

If you specify only the SWAP or NOSWAP option, MOVE is the default assignment.

## Using a Disk-Swap File

You can specify the location of the disk-swap file by using the SWAPPATH configuration command. A disk-swap file is a file that keeps the segments that are temporarily moved out of memory. The SWAPPATH command is used in conjunction with the MEMMAN command. It improves performance to keep the swap file on a separate physical disk from where the majority of your disk activity takes place. For example, if you want MS OS/2 *MULTIUSER* to write the swap file to your second fixed disk (drive D), add the following lines to your *CONFIG.SYS* file:

MEMMAN=SWAP,MOVE

SWAPPATH=D:\

If the SWAPPATH command is not specified in your *CONFIG.SYS* file, MS OS/2 *MULTIUSER* writes the disk-swap file to the root directory of the drive from which your system starts.

**NOTE:** If you are using the SWAPPATH command, make sure your system has ample space available. The minimum size of a disk-swap file is 512K.

## Changing the Number of Disk Buffers

You can change the number of disk buffers (work areas in memory) by using the BUFFERS configuration command. MS OS/2 *MULTIUSER* uses disk buffers as temporary work space while reading and writing data when you use the file allocation table (FAT) file system. Suppose that you want to change the number of buffers to 15. To do this, add the following line to your *CONFIG.SYS* file:

```
BUFFERS=15
```

Each disk buffer is a 512-byte block of memory. You can specify from 1 through 100 buffers. During installation, the value of the BUFFERS command is set to 30. If you remove the BUFFERS command from your *CONFIG.SYS* file, the default value is 3.

## SETTING UP A DEVICE DRIVER

You can set up a device driver by using the DEVICE configuration command. A device driver is a program that allows MS OS/2 *MULTIUSER* to use devices in your system. These devices may include the keyboard, monitor, disk drives, and system clock.

For more information about MS OS/2 *MULTIUSER* device drivers, see Chapter 10, "Using Citrix *MULTIUSER* Device Drivers."



## SETTING UP YOUR ENVIRONMENT VARIABLES

Environment variables are ASCII strings that can be assigned values of your choice. The set of these strings is known as the environment.

To assign a value to an environment variable, use the SET configuration command. To use this command, specify the environment variable name followed by the value that will be assigned to it, as shown below.

```
SET TMP=C:\TMP
```

Environment variables commonly set in your *CONFIG.SYS* file include PATH (directory search path) and DPATH (data search path). Programmers will often set the LIB (library search path), INCLUDE (include-file search path), and TMP (temporary-file directory) environment variables in their *CONFIG.USR* file. You can also define and set your own environment variables.

To set the PATH environment variable so that MS OS/2 *MULTIUSER* searches the current directory first (the default setting), the BIN directory on drive C second, and the OS2 directory on drive C third, add the following line to your *CONFIG.SYS* file:

```
SET PATH=C:\BIN;C:\OS2
```

You can turn on the MS OS/2 *MULTIUSER* command-line editing feature by setting an environment variable. To do this, place the following line in your *CONFIG.SYS* file:

```
SET KEYS=ON
```



## Predefined User Environment Variables

Four predefined environment variables are set up by the system when the user logs in:

- HOME
- USERNAME
- GROUPNAME
- HOSTNAME

The HOME environment variable identifies the user's home directory. The USERNAME and GROUPNAME variables correspond to the user's *loginname* (that is, the *username* and *groupname*, respectively). The HOSTNAME environment variable is set to the system's name.

These variables can be accessed from a batch file by using the % character to delimit the variable name. For example, to use the HOME variable in a batch file, you could use the following command:

```
CD %HOME%
```

This command changes the current directory to your home directory.

## Inheriting Environment Variables

Normally, environment variables such as PATH and DPATH are set up in your *CONFIG.SYS* file. The first time the MS OS/2 *MULTIUSER* command interpreter (CMD) or Program Selector is run, these environment variables are used to set up the initial working environment.

When you start CMD from the Program Selector, the new session inherits the environment variables found in the *CONFIG.SYS* file. If you start CMD from an existing session, however, the new version of CMD inherits the environment variables of the current session. If you have modified any environment variables, the current environment may not be the same as the environment that is specified in the *CONFIG.SYS* file.

## STARTING A PROGRAM WHEN YOU START YOUR SYSTEM

By using configuration commands, you can have MS OS/2 *MULTIUSER* start programs when you start or restart your system.

Use the RUN configuration command to specify a program that you want to start as a background program when you start or restart your system. Suppose that you want to start the program *FASTYPE.EXE* each time you start MS OS/2 *MULTIUSER*. If this program is located in the root directory of drive C, you would add the following line to your *CONFIG.SYS* file:

```
RUN=C:\FASTYPE.EXE
```

Keep the following points in mind when you use the RUN configuration command:

- You can specify multiple RUN commands in your *CONFIG.SYS* file.
- You cannot specify a batch program with the RUN command.

- MS OS/2 *MULTIUSER* processes RUN commands in the order in which they appear in the configuration file (*CONFIG.SYS* or *CONFIG.USR*).
- MS OS/2 *MULTIUSER* processes all DEVICE commands before processing RUN commands.
- Programs specified by the RUN commands are started under a loginname that is determined by which configuration file is used (*CONFIG.SYS* or *CONFIG.USR*).

## SETTING UP CODE PAGE SUPPORT

Code page support is very dependent upon the capabilities of the terminal; thus there are limitations to this support on terminals that have limited hardware capabilities. Code pages are supported fully on the host computer system console display and keyboard. Code page support is limited on all serial port attached terminals. The following section describes the code page support available in MS OS/2 *MULTIUSER*. The two subsequent sections describe the support and limitations that exist while running at the system console and while running at a serial connected terminal.

MS OS/2 *MULTIUSER* provides national-language support through the use of code pages. A code page is a set of 256 characters that MS OS/2 *MULTIUSER* recognizes and processes. MS OS/2 *MULTIUSER* currently supports two code pages: United States (437) and multilingual (850). New programs use the multilingual code page.

You can set up one code page for a single country or you can set up two code pages to enable code-page switching. If you set up two code pages, you can use the CHCP command to switch between them.

## Setting the Country Code

To define country-dependent information such as time, date, and currency conventions, use the *COUNTRY* configuration command. To use this command, type *COUNTRY=* (in the *CONFIG.SYS* file) followed by the country code. If your *COUNTRY.SYS* file is not in the root directory of your start-up drive, type the drive, directory path, and filename after the country code.

For example, if your *COUNTRY.SYS* file is in the *OS2* directory on drive C, you would type the following to specify the country code for the United States:

```
COUNTRY=001,C:\OS2\COUNTRY.SYS
```

The country code 001 sets country specific information for the United States.

The following list shows the United States country code, default code-page assignment (assigned during the installation of MS OS/2 *MULTIUSER*, when you choose the country), keyboard code, and keyboard subcode. The primary code page is listed first, followed by the secondary code page. Use the values in this list when you add the *COUNTRY*, *CODEPAGE*, and *DEVINFO* configuration commands to your *CONFIG.SYS* file. The United States code-page information is set up automatically during installation.

<u>Country</u>	<u>Country Code</u>	<u>Code Pages</u>	<u>Keyboard Code</u>	<u>Keyboard Subcode</u>
United States	001	437,850	US	103



## Preparing Code Pages

To select the code page(s) that will be prepared for use, use the CODEPAGE configuration command. To use this command, type CODEPAGE= (in the *CONFIG.SYS* file) followed by one or two code-page numbers. (If you are preparing two code pages, separate the two numbers with a comma.)

Valid code page numbers are as follows:

<u>Number</u>	<u>Code Page</u>
437	United States
850	Multilingual

For example, to prepare code pages 437 (United States) and 850 (Multilingual), type the following:

```
CODEPAGE=437,850
```

The United States code page information is set up automatically during installation. If you prepare two code pages, you can use the CHCP command to switch back and forth between the two character sets.

## Preparing Devices for Use with Code Pages

### Keyboard

To prepare the console keyboard so that it can use the code pages that you specified with the CODEPAGE configuration command, use the DEVINFO configuration command. If you want to be able to switch code pages, one DEVINFO command is required for each device you prepare. The DEVINFO command for the console keyboard specifies your



keyboard layout and the file that contains the keyboard translation tables. To set up the console keyboard for use with code pages, type `DEVINFO=KBD` followed by a keyboard code and the drive, directory path, and name of the file containing the keyboard translation table. By default, MS OS/2 *MULTIUSER* supplies keyboard translation tables in a file called *KEYBOARD.DCP*.

For example, to prepare a United States keyboard for use with code pages, place the following line in your *CONFIG.SYS* file:

```
DEVINFO=KBD,US,C:\OS2\KEYBOARD.DCP
```

**NOTE:** `DEVINFO` is used for the console keyboard and does not apply to keyboards attached to terminals.

## Screen

The `DEVINFO` command for your console screen specifies the type of screen (either EGA or VGA) and the file that contains a video-font table for displaying characters in each of the supported code pages. To set up the console screen for use with code pages, type `DEVINFO=SCR` followed by the type of display adapter you have and the drive, directory path, and name of the file that contains the system code pages for the screen. The file *VIOTBL.DCP* contains the screen-translation table.

For example, to prepare a console EGA display for use with code pages, place the following line in the *CONFIG.SYS* file:

```
DEVINFO=SCR,EGA,C:\OS2\VIOTBL.DCP
```

**NOTE:** `DEVINFO` is used for the console screen and does not apply to screens attached to terminals.

## Printer

The DEVINFO command for the printer specifies the printer type, the printer name, and the file that contains a printer-font table for each code page supported by MS OS/2 *MULTIUSER*. To set up the printer for use with code pages, type DEVINFO= followed by the port name, the printer name, and the drive, directory path, and name of the file that contains the printer-font tables.

For example, to prepare an IBM 4201 printer that is connected to the LPT1 printer port for use with code pages, add the following line to the *CONFIG.SYS* file:

```
DEVINFO=LPT1,4201,C:\OS2\4201.DCP
```

If your computer includes code-page information in read-only memory (ROM), specify the code page and font-identification number for each ROM or printer cartridge by using the ROM specification after the name of the file containing the printer-font table. If you do not specify multiple fonts or font-identification numbers, the DEVINFO command uses zero as the font-identification number.

For example, to prepare an IBM 4201 printer that is connected to printer port LPT1 for use with code page 437 that is contained in ROM, add the following line to the *CONFIG.SYS* file:

```
DEVINFO=LPT1,4201,C:\OS2\4201.DCP,ROM=(437,0)
```

Remember that you must have one DEVINFO command for each device on your system if you want to be able to switch code pages.

## System Console Code Page Support

Code pages are supported at the console for both keyboard code pages and video code pages fully based upon the parameters that are defined on the *DEVINFO*, the *COUNTRY*, and the *CODEPAGE CONFIG.SYS* commands.

## Terminal Code Page Support

Code-page support for terminals is limited to keyboard code pages. Unlike video code page support, the keyboard code page support requires no underlying hardware to aid in the keystroke translation process; therefore, keyboard code page support is provided at terminals independent of the hardware. When a request is made to change the keyboard code-page, the system will carry out the request and begin translating using the designated code page for the country that is defined on the "COUNTRY =" line within the *CONFIG.SYS* file.

Video code pages are not supported at serial connected terminals; however, if a request is made to change the code-page, the system will honor the request by returning that the request was successful. Further video output that is dependent on the code page may not appear on the screen correctly. The video code page displayed at a terminal is defined by the terminal hardware and is usually not changeable.

## CONTROLLING SYSTEM TRACING

The system trace is a record of actions, such as hardware interrupts or functions, that are taken or processed by the operating system while it runs. Two configuration commands control system tracing: *TRACEBUF* and *TRACE*. (There are



also two utilities you can use to control system tracing: TRACE and TRACEFMT. For information about these utilities, see the *Citrix MULTIUSER Command Reference*.)

## Setting the Size of the System-Trace Buffer

To set the size of the system-trace buffer, place a TRACEBUF command in your *CONFIG.SYS* file. The system-trace buffer is where system-trace events are stored. If you enable tracing but do not specify the size of the trace buffer, a 4K buffer is automatically set up for you. To use the TRACEBUF command, type TRACEBUF= followed by the size of the trace buffer (in kilobytes). The size can be from 1 through 63K. For example, to set up a trace buffer of 8K, add the following line to your *CONFIG.SYS* file:

```
TRACEBUF=8
```

A trace buffer of between 4 and 8K is enough for most trace activities.

## Turning Tracing On or Off

### *CONFIG.SYS*

To turn tracing on or off for some or all events, use the TRACE configuration command. To use this command, you should specify an event code that corresponds to a type of system activity, such as file-system events. The event code must be a decimal number from zero through 255. To turn tracing on, type TRACE=ON followed by one or more event codes. Typing TRACE=ON without an event code records all system-trace events. To turn tracing off, type TRACE=OFF followed by one or more event codes. Typing TRACE=OFF without an event code cancels the recording of all system-trace events.

You can turn tracing on for specified event codes. For example, to turn on only the system-trace events zero and 1, add the following lines to your *CONFIG.SYS* file:

```
TRACE=OFF  
TRACE=ON 0, 1
```

You can also selectively turn tracing off for certain events. If you want to turn tracing on for all events except 31 through 34, add the following lines to your *CONFIG.SYS* file:

```
TRACE=ON  
TRACE=OFF 31, 32, 33, 34
```

You can use the TRACEFMT utility to view trace-event records. For more information about using TRACEFMT, see the *Citrix MULTIUSER Command Reference*.

## TRACE Utility

To turn tracing on or off for some or all events, use the TRACE utility. An example of the TRACE utility follows:

```
TRACE ON 32,33
```

See the command line syntax and the supported options for the TRACE utility in the *Citrix MULTIUSER Command Reference*.

## CONTROLLING SYSTEM LOGGING

The MS OS/2 *MULTIUSER* system logging facility is established with the *CONFIG.SYS* LOG command. During a standard system installation, the logging facility is enabled. The LOG=ON command is placed in the *CONFIG.SYS* file. This command starts the logging process and sets aside a logging buffer in memory and on disk.



System logging is disabled by placing LOG=OFF in the *CONFIG.SYS* file and rebooting. See Chapter 14, "System Event Logging," for more information on the security auditing and event logging established with LOG=ON in the *CONFIG.SYS* file.

## ENABLING AND DISABLING SECURITY ACCESS CHECKS

The MS OS/2 *MULTIUSER* Security System access checking is established via the *CONFIG.SYS* SECURITY command. With SECURITY=ON, all access checking is enabled. Security access checking is ON by default.

Access checking is disabled with SECURITY=OFF in the *CONFIG.SYS* file. See Chapter 7 for more information on security access checking.

## ENABLING AND DISABLING RESOURCE MANAGEMENT

The MS OS/2 *MULTIUSER* resource management facility is established with the *CONFIG.SYS* RESOURCE command. The default RESOURCE=ON command in the *CONFIG.SYS* file causes the system to enforce user limits to ensure a distribution of resources as defined by the System Administrator.

Resource management is disabled by placing RESOURCE=OFF in the *CONFIG.SYS* file. This allows users to compete for resources such as memory, threads, semaphores, and file handles, on a first-come, first-serve basis.

The System Administrator defines user resource limits when configuring user profiles. These limits are used to restrict/protect a user's consumption of the different types of resources. The limits are enforced on a per user basis.

The QUERY LIMITS command is useful for the System Administrator in adjusting user limits.

If RESOURCE=OFF in the *CONFIG.SYS* file, resource limits are not checked by the system.

## CONTROLLING CONSOLE KEYBOARD HOTKEYS

System Installation places a REBOOT=OFF configuration command in *CONFIG.SYS*. This causes the system console keyboard to work exactly as any terminal keyboard.

The *CONFIG.SYS* command REBOOT=ON causes the system console keyboard hotkeys to work as follows:

### ■ Restart Terminal Hotkey

At the system console, if REBOOT=ON is specified in the *CONFIG.SYS* file, **CTRL+ALT+DEL** terminates all applications running on the system and restarts the system. If REBOOT=OFF, the Restart Terminal Hotkey terminates only the console applications and restarts the console terminal.

**NOTE:** The SHUTDOWN command should always be used to terminate all system applications.

### ■ System Dump Hotkey

This hotkey is enabled with the *CONFIG.SYS* command REBOOT=ON and is enabled for the system console only. Press **CTRL+ALT+NUMLOCK** twice to copy the contents of system memory on a disk prepared using the CREATEDD command.

For additional information on the CREATEDD, REBOOT, and SHUTDOWN commands, see the *Citrix MULTIUSER Command Reference*.



# CHAPTER 10

## USING Citrix *MULTIUSER* DEVICE DRIVERS



## INTRODUCTION

A device driver enables MS OS/2 *MULTIUSER* to use a device in your system. MS OS/2 *MULTIUSER* contains various device drivers, including a device driver for the keyboard, screen, printer, floppy disk drive, fixed disk drive, and clock. MS OS/2 *MULTIUSER* automatically loads these drivers during installation.

MS OS/2 *MULTIUSER* also contains device drivers that you use with some optional devices. Optional devices include a serial communications port or an external floppy-disk drive. The MS OS/2 *MULTIUSER* Installation process sets up *CONFIG.SYS* to load certain optional device drivers. If there is no statement in *CONFIG.SYS* to load an optional device driver after system installation, you need to set up the drivers for these optional devices, since MS OS/2 *MULTIUSER* does not load them automatically.

Additional MS OS/2 *MULTIUSER* device drivers work with your system to support or enhance its capabilities. These include the virtual disk device drivers. These drivers also need to be set up.

In addition to the device drivers MS OS/2 *MULTIUSER* offers, device drivers are generally provided with any devices you might purchase.

For more information about using a specific device driver, see the entry for that driver in the *Citrix MULTIUSER Command Reference*.



## SETTING UP AN MS OS/2 *MULTIUSER* DEVICE DRIVER

You set up a device driver in MS OS/2 *MULTIUSER* by using the **DEVICE** configuration command to specify the name of the driver to load and its location in your system. The **DEVICE** command is described in Chapter 9, "Using System Configuration Commands." In general, if a device driver is not located in the root directory of the disk from which you start your system, you need to specify the full path of the device driver in the **DEVICE** command. You can set up a device driver in your system when you install MS OS/2 *MULTIUSER*, or by adding **DEVICE=** configuration commands to *CONFIG.SYS*.

You may also be able to use the **DDINSTAL** utility to install a device driver if the device driver is setup to be installed with **DDINSTAL**. **DDINSTAL** copies device driver files to your system and adds the appropriate **DEVICE** configuration command to your *CONFIG.SYS* file.

**DDINSTAL** sometimes requires the system to be booted from the Install diskette during its processing. Because of this you should run **DDINSTAL** while in maintenance mode. To use the **DDINSTAL** utility, follow these steps:

1. At the MS OS/2 *MULTIUSER* command prompt, type **DDINSTAL** and press **ENTER**.
2. Insert the disk that contains the device driver to be installed and press **ENTER**.
3. **DDINSTAL** lists any **.DDP** (device-driver profile) files it finds. Select the device driver you want set up from the list and press **ENTER**. (You can select more than one driver.)

4. When prompted, insert the MS OS/2 *MULTIUSER* Install disk and restart your system by pressing **CTRL+ALT+DEL**. (Although you use the Install disk, you will not go through the complete installation process again.) When the system restarts, the DDINSTAL utility installs the device driver.
5. When the program is finished, remove the Install disk and restart your system.

## CHANGING A DISPLAY DEVICE DRIVER

During installation, MS OS/2 *MULTIUSER* determines what kind of display you have and loads the appropriate display device driver. If you replace your display, you may need to change the display driver. The following section describes how to change a display driver without running the installation program again. To change the display driver, follow these steps:

1. Define the VIDEO\_DEVICES environment variable by using the SET configuration command in your *CONFIG.SYS* file and listing one or more environment variables that describe video device handlers. For example, to use an IBM Enhanced Graphics Adapter, define the VIDEO\_DEVICES environment variable by placing the following line in your *CONFIG.SYS* file:  
  
SET VIDEO\_DEVICES=VIO\_IBMVGA
2. Define the environment variable that you specified with the set VIDEO\_DEVICES command by specifying a video device handler.

Each video device handler is made up of one or more components that provide a set of functions to be used by a specific display adapter.

Using the example from step one of the preceding list, you would define the `VIO_IBMEGA` variable by specifying the `BVHEGA.DLL` dynamic-link library, placing the following line in your *CONFIG.SYS* file:

```
SET VIO_IBMEGA=DEVICE(BVHEGA)
```

As another example, suppose you were using an IBM 8514 display attached to an IBM 8514A display adapter. You would need to place the following lines in your *CONFIG.SYS* file:

```
SET VIDEO_DEVICES=VIO_IBM8514A  
SET VIO_IBM8514A=DEVICE(BVHVGA,BVH8514A)
```

In the preceding example, the `BVHVGA.DLL` and `BVH8514A.DLL` dynamic-link libraries both support the IBM 8514A display adapter.

You can also specify the use of two displays. The following example shows the configuration commands you would specify if you were using both an IBM 8514 display attached to an IBM 8514A display adapter and an IBM PS/2 display attached to a VGA adapter:

```
SET VIDEO_DEVICES=VIO_IBMVGA,VIO_IBM8514A  
SET VIO_IBMVGA=DEVICE(BVHVGA)  
SET VIO_IBM8514A=DEVICE(BVH8514A)
```

## USING A SERIAL COMMUNICATIONS PORT

You should not use a port as a general communications port if you are using it for a terminal. You can use the serial communications port by setting up the appropriate device driver for it.

Before you set up a device driver for a communications port, you must choose the driver that works with your hardware. Refer to the following list to find the driver that matches your hardware:

<u>Driver</u>	<u>Hardware</u>
COM01.SYS	Industry Standard Architecture (ISA) 386 compatible computers
COM02.SYS	Micro Channel Architecture 386 compatible computers

To direct MS OS/2 *MULTIUSER* to load the *COM01.SYS* device driver from the OS2 directory on drive C, add the following line to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\COM01.SYS
```

When this communications-port driver is loaded, MS OS/2 *MULTIUSER* can use the serial communications ports on your Industry Standard Architecture 386 computer.

To use a communications port, you need to set up the port in addition to specifying the device driver if the system is not using the port as a terminal. Set up a communications port by using the MODE utility. For information on the MODE utility, see "System Operation."

MS OS/2 *MULTIUSER* Installation will ensure the appropriate communications port driver is installed.



## USING A LOGICAL DRIVE LETTER

You can use a logical drive letter to name a floppy-disk drive by setting up the *EXTDSKDD.SYS* device driver. A logical drive letter, in this case, is a name for a physical disk drive. (The logical disk drives that you can set up in an extended partition on your fixed disk are also parts of a physical disk drive, but they are different from the logical drives that you use with *EXTDSKDD.SYS*.) Using a logical drive letter, you can name an external disk drive or you can assign a second name (an alias) to a disk drive and copy files to different places on the same disk drive. For example, to direct MS OS/2 *MULTIUSER* to load the *EXTDSKDD.SYS* device driver from the OS2 directory on drive C, add the following line to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:2
```

When MS OS/2 *MULTIUSER* loads the driver, your MS OS/2 *MULTIUSER* programs can use an IBM external 720K 3½-inch floppy-disk drive. By default, this logical drive has 80 tracks, nine sectors per track, and two read/write heads. The next available drive letter is assigned to the logical drive.

If you want to copy from a logical drive on the external disk drive to a different logical drive on the same disk, add the following lines to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\EXTDSKDD.SYS /D:2  
DEVICE=C:\OS2\EXTDSKDD.SYS /D:2
```

This example directs MS OS/2 *MULTIUSER* to load the *EXTDSKDD.SYS* device driver. The first line associates the next available drive letter with the external disk drive. The second line associates an additional drive letter (an alias)



with the same external drive. This allows you to copy data from the external disk drive to a logical drive on the same disk.

## USING A VIRTUAL DISK DRIVE

You can use a virtual disk drive by setting up the *VDISK.SYS* device driver. When you use a virtual disk drive, your system simulates a disk drive in memory and provides quick access to files stored there. You may want to use a virtual disk to store temporary files. Since temporary files are deleted when they are no longer needed, there is no reason to save them on a disk. To direct MS OS/2 *MULTIUSER* to load the *VDISK.SYS* device driver from the OS2 directory on drive C, add the following line to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\VDISK.SYS
```

This line must come after any *DEVICE* commands that load the *EXTDSKDD.SYS* device driver in your *CONFIG.SYS* file.

If the driver is found, MS OS/2 *MULTIUSER* loads it and creates a 64K virtual disk with 128-byte sectors, 64 directory entries, and the name of the next available drive letter. (These are the default values.)

Any information stored on a virtual disk is lost when your computer is restarted or turned off. If you use a virtual disk for temporary files, be sure to specify the appropriate drive letter with the *SET TEMP* configuration command in your *CONFIG.SYS* file. This command tells MS OS/2 *MULTIUSER* where to place all temporary files.

You can have more than one virtual disk drive. Drive letters for virtual disks are assigned in the order in which the *DEVICE* commands appear in the *CONFIG.SYS* file. When you specify memory for a virtual disk, that memory is no longer available for other programs to use.

You need not format a virtual disk drive. It is automatically configured as a FAT file system drive.

You may want to increase the size of the virtual disk drive, since 64K is not very much space. You can do this by adding the following line to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\VDISK.SYS 128
```

This example creates a 128K virtual disk with 128-byte sectors and 64 directory entries.

You may also want to change the number of directory entries for the virtual disk drive. Do this by adding the following line to your *CONFIG.SYS* file:

```
DEVICE=C:\OS2\VDISK.SYS 256,,128
```

This example creates a 256K virtual disk with 128-byte sectors and 128 directory entries. Note that a sector size is not specified; the place where it would be specified is designated by a comma. (If you wanted to specify all three values, you would separate them with spaces.)

When you specify a value for the number of directory entries, the value is rounded up to the nearest sector-size boundary. For example, if you give a value of 25 and your sector size is 512 bytes, 25 will be rounded up to 32, which is the next multiple of 16. (There are sixteen 32-byte directory entries in 512 bytes.)

The maximum size of a virtual disk depends on the amount of available memory in your system but it cannot be larger than 4 megabytes. If the virtual disk size specified is too large to fit in memory, *VDISK.SYS* will attempt to make a virtual disk that is 16K. This may result in a virtual disk with a different number of directory entries than you specified.

When you start or restart your system, MS OS/2 *MULTIUSER* displays information about any virtual disk it creates. The information includes the assigned drive letter, the size of the virtual disk, its sector size, and the number of directory entries. You can also use the CHKDSK utility to determine the size of a virtual disk.

# CHAPTER 11

## CONFIGURING APPLICATIONS

## INTRODUCTION

Even though most applications written for MS OS/2 *MULTIUSER* are intended to be used by a single user, most of these applications can be used by multiple users simultaneously without a user being aware that other users are running the same application at the same time.

It is important for you to install your applications in a manner consistent with your security objectives. This may require ensuring that users can execute the application but cannot write into its directory. Some applications may not run if they cannot write into their own directory until some extra work is done to install the application differently than the default installation procedure for that application.

Some applications may keep their state information in files between invocations of the program. It is important to remember that all programs running in the system see the file system file names the same way regardless of which user starts the program. The only difference may be what the default directory is when the program is started.

Additional information on configuring specific applications is covered in the *README* files. See Appendix A.

## DIRECTORY STRUCTURE AND SECURITY CONSIDERATIONS

Security adds a dimension to the installation of an application. In an unsecured system all the programs, files, and devices used by an application are always accessible to all who use it. For a secured system, different users have different views of the file system; consequently, you must allow for the ability of each set of application users to get to



the resources the application needs. Also, depending on your installation strategy for security and system integrity, you may be concerned with preventing a user from gaining access to restricted files.

This section describes, in general terms, the types of things you should consider and understand when installing a new application.

## Categorizing Applications

Applications come in a variety of styles. The style dictates how you will need to set up access to the data and the programs associated with the application.

A common style is a single user MS OS/2 multitasking application. This simply means that the application will run in multiple MS OS/2 *MULTIUSER* sessions without causing conflict of data files between the two sessions, either because the data files are generated by the user, or because the application recognizes and avoids conflicts with its work files. You can usually install one copy of this type of application and have several users share it simultaneously.

**NOTE:** Check your application to make sure your usage conforms to application licensing requirements.

Another style is a Local Area Network(LAN)-based product that uses shared files to communicate. These often behave the same as the single-user MS OS/2 application. However, the application may depend on network services; in this case, the application cannot be installed on standalone MS OS/2 *MULTIUSER*. Often, however, these types of LAN-based applications simply do a series of NET USE and NET SHARE operations to gain accessibility to drives. These will not be needed in standalone MS OS/2 *MULTIUSER*.

LAN-based products that use a client-server model have further implications on network services and on the environments in which they are installed in MS OS/2 *MULTIUSER*. From a file system perspective, they can be treated like the others already described.

A few MS OS/2 applications or utilities were originally written for a single-task, single-user environment. These do not tolerate multiple copies of the same program running simultaneously in different sessions. You may have to install multiple copies of this type of application so that each user has a copy. This section does not address further considerations of this type of application.

## When Installing an Application

Each application you install may be different. One will set up its own directory structure in a very specific place; another will allow you to specify the drive letter on which you wish to place the files; still another will allow you to indicate any subdirectory.

Some applications change the system *CONFIG.SYS* file and possibly the system *STARTUP.CMD*.

It is important to understand what the application is meant to do before following its install process so you can ensure that you have proper access to all target files and directories during install. In most cases, in an installation using the standard security layout presented in this guide, an Administrator will have proper access to the system to complete an application installation.

In general, an application install process does not require that you be in maintenance mode; however, a complex application that has modified *CONFIG.SYS* will require you to restart the system to complete the installation. In this case,

you should follow the SHUTDOWN procedures described in this guide regardless of the instructions in the application documentation. Remember, many applications are written with a single-user environment in mind.

## **CONFIG.SYS and Security**

As mentioned above, an application may require modification to the *CONFIG.SYS* file. You should take note of the changes being made to *CONFIG.SYS*; they may not fit in with your security strategy and you may have to make subsequent modifications to your security configuration or to *CONFIG.SYS*.

These are the typical changes to *CONFIG.SYS* and what considerations you may have to deal with:

- Add its *.DLL* path to LIBPATH  
*.DLL* (dynamalink library) paths will require that the user or program using the path have Execute access. An extensive discussion of this is found in Chapter 7, "Configuring Security."
- Add its programs and data paths to SET PATH and SET DPATH  
A similar consideration exists for PATH as for LIBPATH. Usually, it is simply a matter of ensuring that users of the application have Execute access to these paths. For DPATH, access will depend on the files and the application. This is discussed further below.

PATH and DPATH as set in *CONFIG.SYS* will not necessarily "take hold" for a user; the user's *CONFIG.USR* or other SET PATH statement could nullify what is put in *CONFIG.SYS*.



■ Add device drivers

Note that device drivers usually add a character device name (or a set of names) to the system. These may be new devices you may wish to secure.

Note also that device drivers are essentially extensions of the MS OS/2 *MULTIUSER* operating system; as such, they have complete access to all facilities they need. The important security consideration here is to know and understand the purpose of the device driver. If you are concerned with a serious security breach, you must ensure that the device driver you have installed will serve its legitimate purpose.

■ Enable IOPL

Some applications require IOPL capability and may automatically change your *CONFIG.SYS* IOPL statement to IOPL=YES. This creates a potential security risk. If this happens, you may wish to change the IOPL statement to a list format. A complete discussion of IOPL is found in Chapter 9, "Enabling Input/Output Privilege Levels."

## Setting up Users for the Installed Application

Once you have installed the application, you need to set up the program and data paths of the application so that users can access them. In some cases, this is a simple matter of giving Read (R) and Execute (X) access to a user. But, many applications have a more complex scheme for organizing programs and data. To properly set up user environments, you must understand the basic structure of the application's file and directory layout.

## Access to the Programs

A user must have a minimum of Execute (X) access to run a program. X access will allow the user to run the programs but not copy them. For most applications, it is easiest to give the user or user group X access to the directory in which the programs were placed.

You may want to restrict access to some applications that have maintenance-only type programs. For these, you can selectively change the security access. There are also cases where one application program runs another. The user must have Execute access to both of these or the program that runs the other program must be registered and given access.

Assuming that the user has execute access to the programs, the PATH environment variable can be set up via *CONFIG.SYS*, *STARTUP.CMD*, or *CONFIGUSR*.

## Access to the Data Paths

The requirements here are less predictable because data management is very different for each application. This section outlines several types of files you may encounter, and suggestions on how to deal with them.

### ■ User generated data files

These files are created on behalf of the user and apply only to that user. Applications usually allow the user to specify the data path for this type of file. A user can indicate that the file is to be in a personal directory; one to which the user has Read, Write, Create, and Delete permissions.

An example of this type of file is a word processing document.



### ■ User oriented application setup files

Often an application will use a setup file that contains a default setting to be used when the application starts. These files can usually be modified by a user through options in the application itself. A setup file is customized to a user; one user should not get the setup defaults of another.

Much of the success of dealing with these files depends on the application. One common way is for the application to get the setup file from the current directory or from DPATH. In the current directory case, it is only important that the user start the program from the directory where the setup file resides. Depending on how you have configured the user environment and terminal environment, you may need to establish the proper user working directory with the CONFIG command. For the DPATH case, you or the user need to ensure that the DPATH environment variable is properly established.

Applications sometime find a setup file by requiring that the file or path name be entered when the application starts, say as a command line option. If the program is invoked from the Program Selector or from the *CMD.EXE* command line, simply include this parameter. If you have set up the user's first program as the application, you will need to include this parameter in the user's profile.

### ■ Application work files

These files are created and updated by the application in the normal course of execution. The application must have Read, Write, Create, Delete and possibly other permissions to the directory in

which this file is created. This is accomplished either by giving the user this level of access or by registering the program and giving the program alias this access.

These may be more difficult to deal with. One type of work file is a temporary file that is kept in the application directory. In this case, if you have given users complete access to the application directory, a user can read, change, or delete data that may be associated with another user. If this violates your security plan, you should register the application and give its alias a different level of access than the user.

## SYSTEM GLOBAL/USER LOGIN GLOBAL CONSIDERATIONS FOR APPLICATIONS

Resource Management separates certain shared resources into user login specific contexts. This prevents conflicts for resources when multiple users attempt to run the same application. As an example, if a word processor and spreadsheet use named shared memory for a cut-and-paste buffer, each user must have a different cut-and-paste buffer for his or her work. If they did not, then one user would be able to cut-and-paste another user's data. This separation of shared resources actually applies to each user login. If a user logs into the system using the same *loginname*, his cut-and-paste buffer will be unique to that login. Note that this user login separation does not apply to file system file names.

For some applications (client/server multi-user enabled), a system global context may be needed instead of a user login context. For these programs, the REGISTER utility is used to mark the appropriate *.EXE* or *.DLL* file(s) as System Global.

Use of shared resources should be kept in mind when configuring applications. Most applications written for MS OS/2 were designed for single-user systems. These applications should run unchanged on MS OS/2 *MULTIUSER* since Resource Management ensures separate user login contexts for each invocation. Some applications have already been designed for a multiuser environment. These applications may have to be marked "System Global" via the REGISTER utility.

## When to Use the REGISTER Utility

There are few applications that should require System Global resources. These applications will be designed for a multiuser environment (for example, Client/Server applications). Run the application. If it does not work correctly, some or all of its *.EXE* and *.DLL* files may need to be marked System Global.

Unless you are familiar with the application's software, it may not be obvious whether marking part or all of the application will solve the problem. If marking all of the executable files does not solve the problem, try marking different combinations of the application's executable files. For example, for a Client/Server application try marking all the server executable files and only the client's *.DLL* files.

## How to Use the REGISTER Utility

Use the REGISTER utility to query or set an application's resource context. If the SYSTEM parameter is specified, the application/dynamic link library will be marked System Global. If the USER parameter is specified, the application/dynamic link library will be marked User Login Global. Below are listed a few examples of the REGISTER utility. For more information on the REGISTER utility see the *Citrix MULTIUSER Command Reference*.



REGISTER MYEXE.EXE /ALIAS:SpecialPrograms /SYSTEM  
Sets *MYEXE.EXE* to SYSTEM GLOBAL, also registers this program under the SpecialPrograms alias.

REGISTER MYDLL.DLL /USER  
Sets *MYDLL.DLL* to USER GLOBAL. In this example, *MYDLL.DLL* had already been registered under an *aliasname*.

REGISTER MYDLL.DLL /Q  
Query the context of *MYDLL.DLL*

The default context for all applications and dynamic link libraries is User Login Global.

## TIME CRITICAL PRIORITY CONSIDERATIONS

MS OS/2 *MULTIUSER* allows applications to set the priorities of the threads that they control. The highest priority value is Time Critical. This priority should only be used by programs that need to run immediately in order to prevent data loss. Typically, this would be a high speed communication program. This priority value is higher than even programs that have been CPU starved (that is, their priority was increased since they have not run in MAXWAIT seconds). For more information on MAXWAIT, see the *Citrix MULTIUSER Command Reference*.

In a multiuser environment, this Time Critical Priority can cause problems:

- If too many applications run time critical threads, then normal user response may degrade drastically.
- An errant application could run out of control consuming all the CPU cycles and lock users out of the system.

In order to prevent these problems, a program must first be registered to gain Time Critical Priority. This is done using the REGISTER and CONFIG ACCESS utilities. If the program is not registered (this is the default) and requests Time Critical Priority, it will receive a priority less than starved programs but above all normal priorities.

## **When to REGISTER a Program for SET\_TIME\_CRITICAL Access**

A possible scenario for detecting which programs use Time Critical Priority is described below:

After installing a program (that appears to possibly require Time Critical Priority), it is recommended to first try the program without registering it for Time Critical Priority access. The EVENTS utility can then be used to see if access was denied on a request for Time Critical Priority. If this occurs, then you can give the program access to the Time Critical Priority and see if the performance improves or degrades. An example of how to use the EVENTS utility is described below:

```
EVENTS /MON:3 /D+:X
```

This example will monitor access denied errors under the Security Audit event. It checks every 3 seconds to see the latest data. If the program is now started and tries to set one of its threads to Time Critical Priority, the event will be displayed by the EVENTS utility. For more information on the EVENTS utility, see the *Citrix MULTIUSER Command Reference*.

## **How to REGISTER a Program for SET\_TIME\_CRITICAL Access**

The REGISTER utility is used to give the program an aliasname that will have special security authorizations (that is, Time Critical Priority). The CONFIG ACCESS utility is then used to grant the aliasname access to the Program Function Point SET\_TIME\_CRITICAL. Now the program will be able



to set its priority to Time Critical. The default for all programs is to NOT have the security authorization to make themselves Time Critical. An example is described below:

```
REGISTER \HISPEED\COMM /ALIAS:TCP
```

This will register the entire \HISPEED\COMM directory under the TCP alias.

```
. CONFIG ACCESS \PFP\SET_TIME_CRITICAL.54 TCP X
```

This will allow all of the programs in the TCP alias access to the Time Critical Priority. The full screen CONFIG ACCESS utility can also be used to configure access to the Time Critical Priority.



# PART 4

## SYSTEM OPERATION AND MAINTENANCE

# CHAPTER 12

## SYSTEM OPERATION

## HANDLING LOGIN AND LOGOUT

### Introduction

The login procedure is a fundamental administrative tool. Its purpose is to block unauthorized access to the system through a terminal. It is the standard program associated with an active but unused terminal. If you wish to bypass the login prompt at a terminal, you should configure the terminal for autologin. The system will then automatically log the specified *loginname* into the system when the autologin terminal is powered on.

### Logging In As The Administrator

The login program is the first program that runs at a configured terminal. When MS OS/2 *MULTIUSER* is installed, the console is automatically configured with a default terminal profile that permits the login prompt to appear at the console the first time the system is started after installation. The login prompt requests the user's *loginname*. To perform system administration when you log into the system, you must use a *loginname* that belongs to the Administrator security class. Installation of MS OS/2 *MULTIUSER* automatically creates a user profile for a System Administrator during the Install Users portion. The *loginname* of this Administrator level user defaults to ADMIN.WRKGRP with a default *password* of "citrix". The initial user program of this *loginname* defaults to the Program Selector. You may choose to change these defaults during system installation. After system installation, you should log into the system with this *loginname* and change its *password* using the PASSWORD utility.

After logging in as a System Administrator, you may choose to create additional user profiles (and *loginnames*) with the Administrator security class. Refer to Chapter 5, "Configuring Profiles" for information on configuring user profiles and to Chapter 7, "Configuring Security," for information on security.

## Login Example

The following prompt appears at the console or at a terminal that is powered on.

Login:

Type the *username* ADMIN. The system sets the *loginname* to ADMIN.WRKGRP by using the default *groupname* for the specified *username*. (The *username* ADMIN may have been changed during the system installation.) Entering the *username* ADMIN results in the *password* prompt.

Login: ADMIN

Password:

Type the *password*, which is initially set to "citrix" for the ADMIN.WRKGRP loginname (but may have been changed during the system initialization). The password you type will not be echoed to the screen but will be represented by underscores ( \_ ) for each character in the *password*.

Login: ADMIN

Password: \_ \_ \_ \_ \_

The initial program appears. For the *loginname* ADMIN.WRKGRP, this program defaults to the Program Selector.



## Changing the Password Example

To change the *password* of your *loginname*, use the PASSWORD utility. The following example demonstrates how to use the PASSWORD utility from the CMD command prompt, changing the *password* of the Administrator *loginname* ADMIN.WRKGRP from "citrix" to "abdmmp". Remember that passwords are case-sensitive. Type the following (on one line):

```
PASSWORD ADMIN.WRKGRP  
/PASSWORD:citrix/abdmmp/abdmmp
```

When you change your *password*, select a new *password* that cannot easily be uncovered by someone else.

## Logging Out

There are two basic methods to terminate your login. The recommended method is to switch to each program running in the login and end the program according to the program's instructions. For the command interpreter, CMD, type EXIT at the command prompt. After the last program in a login is terminated, the login will automatically be terminated by the system. If you are using the Program Selector, you must use the Program Selector logout function after terminating all your applications. This method is preferred because it allows each program to terminate without losing any data.

Another method to terminate your login is to use the LOGOUT program. The LOGOUT program forces all programs in the login to terminate. However, because the program is being forced to terminate, it may lose data.

## Logout Examples

1. Use the **ALT+ESC** hotkey sequence to switch to each session and terminate each program. Use the Program Selector logout function if required.

2. Use the LOGOUT program by typing the following at the command line:

LOGOUT

This will force all programs in the current login to terminate.

## Handling User Logins

After system installation, you configure users with the CONFIG program. This program permits you to grant other individuals access to the system. Each user must be assigned a *username* and *groupname*, which make up the *loginname*. In addition, you may assign a *password* to each *loginname*. This is desirable if you need to protect the access of users to the system. Because *loginnames* of authorized users are well-known to other individuals seeking access to the system, *passwords* ensure protection. Of course, each user is responsible for protecting his or her *password* from becoming known to others.

Refer to Chapter 5, "Configuring Profiles," for more information about user and group profiles.

## Managing the Licensed Number of Users

MS OS/2 *MULTIUSER* sets a limit on the licensed number of users who may use the system. This licensed number of users for the system is based on the number of concurrent connected logins. When a user logs in, the system creates a login for that user which is connected to the terminal where the user is located. When the maximum number of connected logins is reached, the system will not allow users to login. A user who attempts to login at this time will see an error message from the login attempt:

```
LOGIN failed. The maximum licensed login count  
has been reached on this system. Contact your  
System Administrator for assistance.
```

This message indicates that the number of users who want to use the system concurrently has exceeded the licensed number of users.

MS OS/2 *MULTIUSER* sets the licensed number of users to a predetermined value. This number may be increased with the installation of user add-on packs.

To find out the current licensed number of users on your system, use the QUERY HOST command with the /LICENSE option. To determine the number and the identities of users currently logged in, use the QUERY USER command. To force any of the current logins off the system, use the LOGOUT command. Refer to the *Citrix MULTIUSER Command Reference*.

To increase the licensed number of users on your system, you should get a user add-on pack from the supplier of your MS OS/2 *MULTIUSER* and install it.

## Autologin for Users

As you configure users and terminals, you may want to assign certain users to certain terminals. In this case, you may set the autologin parameter in the terminal profile to the *loginname* of the user who will be located at that terminal. When the terminal is powered up, the *loginname* will be automatically logged into the system and the initial program for that *loginname* will be started at the terminal.

Refer to Chapter 6, "Configuring Terminals," for more information on setting up a terminal for an autologin.

## Handling User Passwords

You can assign a *password* to a user when you configure the user profile by selecting the "Option" menu in the Action bar and then selecting the option "Changing Password". Refer to Chapter 5, "User Profiles" for information on setting up *loginnames*. You may assign *passwords* to users after you have configured their user profiles by either editing their user profiles or by using the PASSWORD utility.

If a user forgets his or her *password*, you can give the user a new *password* by editing the user profile. The user should then change this password with the PASSWORD utility.

## Logging Users Out

Should a problem occur in the login of another user and you have no other recourse but to terminate the login, use the LOGOUT command. Be aware that programs running in the user login could lose data.

Should a problem occur with a particular application and you must terminate the application, use the KILL command.

## User Logout Examples

In this example, LOGOUT terminates all logins with ZEUS.WRKGRP as the *loginname*.

```
LOGOUT \ZEUS.WRKGRP
```

In the next example, LOGOUT terminates loginID 23 belonging to the *loginname* of ZEUS.WRKGRP.

```
LOGOUT \ZEUS.WRKGRP\23
```



## SHUTTING DOWN THE SYSTEM

One of the tasks a System Operator and/or Administrator needs to perform is to be able to shutdown the system in an orderly fashion. The MS OS/2 *MULTIUSER* tool that is used to accomplish this is the SHUTDOWN utility. SHUTDOWN is used to prepare the system for power off or any other occasion where the system must be made unavailable for normal use.

The SHUTDOWN utility supports several options that allow the following functions to be performed:

- Shutdown the system.
- Shutdown the system, followed by an immediate reboot.
- Shutdown the system and enter maintenance mode.

It is important that you follow this shutdown procedure rather than just turning off the system or doing a **CTRL+ALT+DEL**. When using HPFS or any other file system, there can be pending disk updates. The shutdown process allows the system to complete outstanding writes and confirm that the disk is in a stable state. Also, remember that there are other users. Shutdown notifies all users and gives them a chance to save their work.

For more information on the SHUTDOWN utility, see the *Citrix MULTIUSER Command Reference*.

## USING FILES AND DIRECTORIES IN HPFS

In HPFS, you can give files and directories more descriptive names than you can in the FAT file system. When using these HPFS naming features, you need to be aware of how HPFS names work with MS OS/2 *MULTIUSER*. Naming rules and the interaction of HPFS filenames with MS OS/2 *MULTIUSER* are described in the following sections.

### Naming HPFS Files and Directories

HPFS file and directory names may include several features that are not available in the FAT file system:

- File and directory names can be up to 254 characters long. (In the FAT file system, there is a limit of 8-character *filenames* and 3-character *filename* extensions.) Paths and *filenames* together can be up to 259 characters long.
- Blank spaces and periods (.) can occur anywhere in the file or directory name. However, blanks and/or periods that occur at the end of a name are not treated by MS OS/2 *MULTIUSER* as a significant part of the name. For example, the filenames "xyz", "xyz.", "xyz. ", and "xyz ." are stored by MS OS/2 *MULTIUSER* as "xyz".
- The following characters can be used in naming HPFS files and directories:

, + = [ ] ;

Note that the following characters are not currently allowed in any file system with MS OS/2 *MULTIUSER*:

< > : " / \ | \* ?

- You can use uppercase, lowercase, or mixed case when naming HPFS files and directories; the name is displayed in the directory list just as you typed it. However, MS OS/2 *MULTIUSER* ignores case in comparing file and directory names. For example, "*Taxfile*", "*TAXFILE*", and "*taxfile*" are the same filename to MS OS/2 *MULTIUSER*; only one can exist in a given directory.

## Using HPFS File and Directory Names with Commands

You must enclose any blanks or special characters in double quotation marks when you type an HPFS filename or directory name on the command line so that MS OS/2 *MULTIUSER* will recognize that they are part of the name. You can put the double quotation marks around the entire name or around just the blanks and special characters. For example, to copy a file named "*My Tax File, 1988.txt*" to a directory on drive C named "Current Taxes", you can type either of the following:

```
COPY "My Tax File, 1988.txt" C:"Current Taxes"  
COPY My "Tax" "File", "1988.txt" C:Current "Taxes"
```

You must also use double quotation marks any time a file or directory that uses the HPFS naming features is listed in batch programs.

## Transferring HPFS Files and Directories

Because not all file systems support the use of HPFS naming features, you must be careful when copying and moving files between file systems. In general, HPFS file and directory names that use HPFS naming features cannot be transferred from HPFS to the FAT file system. (If you use the COPY or MOVE command at the command prompt, you must rename the file; the HPFS filename is lost.)

**NOTE:** In HPFS, you can change the case of a file or directory name (for example, from uppercase to lowercase) by using the MOVE or the RENAME command and giving the file or directory a new name. For example, to change *MYTAXFILE.TXT* to lowercase from the command line, you would type the following:

```
MOVE MYTAXFILE.TXT mytaxfile.txt
```

**NOTE:** If you copy files from HPFS to MS DOS or a version of MS OS/2 that does not support extended attributes (versions earlier than 1.2), HPFS extended attributes are lost.

## Using Wildcard Characters in HPFS

You can use the asterisk (\*) and question mark (?) wildcard characters in HPFS. Wildcard characters take the place of other characters when you specify filenames and directories in commands and utilities and are especially useful when you work with groups of files.

Generally, the asterisk matches zero or more characters in a *filename*. For example, suppose you have the following files:

*M*  
*MAY*  
*MAY.TXT*  
*MAY.TXT.BAK*

If you type *MAY\** when you are using the CHDIR, DEL, DIR, or RMDIR commands in HPFS, the files *MAY*, *MAY.TXT*, and *MAY.TXT.BAK* are matched. If you type *\*.* instead, the files *M* and *MAY* are matched.



When you use the COPY, MOVE, or RENAME commands in HPFS, the asterisk works a little differently. With each of these commands, you specify source and destination *filenames*. When you use the asterisk in the source *filename*, it works as previously described. However, when you use the asterisk in the destination *filename*, the system copies the characters that follow the asterisk to the destination *filename*. For example, if you type COPY \*.C \*.BAK, the COPY command first finds all files that end with .C. Then it makes a copy of each file, using as the name of the new file the characters that precede the .C and adding the new .BAK extension.

The question mark matches any single character in a *filename* except a period (.). For example, *JUNE?* matches *JUNE*, *JUNE1*, and *JUNE2*, but does not match *JUNE3.TXT* or *JUNE.TXT*.

Wildcard characters are valid only in the *filename* that follows the last backslash (\) of a path.

## BACKING UP AND RESTORING FILES AND DIRECTORIES

MS OS/2 *MULTIUSER* provides utilities to allow you to back up files on a diskette and selectively restore them at a later date.

### **Making Backup Copies of Your Files**

The BACKUP utility creates a backup copy of one or more files on a disk and stores the copy on another disk. It provides an easy way to make backup copies of files, directories, or entire disks. Using BACKUP differs from using the COPY command or the XCOPY utility to copy files.

When you use BACKUP to copy files, the contents of the files are stored in a single file instead of in individual files. In addition, BACKUP has several options that let you make backup copies of specific files.

The BACKUP utility is most commonly used to make backup copies of files that are on your fixed disk and store them on a series of floppy disks. You can use these floppy disks to recover your files if the fixed disk becomes corrupted.

You can back up one floppy disk to another floppy disk even if the disks have a different number of sides or sectors.

The BACKUP utility does not make copies of files if the files are already open or if they are in use by another program. For example, if you run BACKUP from your start-up disk, files such as the *SWAPPER.DAT* file are not backed up because they are being used by MS OS/2 *MULTIUSER*. See Chapter 13, "System Maintenance," for instructions on backing up these files.

**NOTE:** The BACKUP utility does not make backup copies of the file *CMD.EXE*, or of hidden system files.

## Making Backup Files on Another Disk

To run the BACKUP utility, type BACKUP followed by the drive that contains the disk you want to back up (the source disk) and the drive that contains the disk that will hold the backup files (the backup disk).

To back up the contents of your current directory to a directory in another drive, specify the drive letters of the source drive and the backup drive. For example, to back up the root directory of your fixed disk (drive C) to a floppy disk in drive A, type the following:

BACKUP C: A:

To make a backup copy of the contents of a specified directory, type the drive letter and path of the directory you want to copy. For example, to back up the files in the ORDERS subdirectory of the ACCOUNT directory to a floppy disk in drive A, type the following:

```
BACKUP C:\ACCOUNT\ORDERS A:
```

Suppose you have several files on your fixed disk that have the *filename* extension *.OLD*. To back up these files to a floppy disk in drive B, use a wildcard character, as shown in the following:

```
BACKUP C:\ACCOUNT\ORDERS\*.OLD B:
```

By default, any files on the backup floppy disk are erased before the backup files are added to it. If all of the files cannot fit on one floppy disk, the BACKUP utility prompts you to insert another backup disk when the current disk is filled.

When the backup operation is complete and you look at a directory list for the backup disk, you will not see the names of the files you just copied. Instead, you will see two files named *BACKUP.nnn* and *CONTROL.nnn*. These files contain your files and paths; they will be used later to restore these files. On the first backup disk, these two files will be called *BACKUP.001* and *CONTROL.001*; on the second disk, they will be called *BACKUP.002* and *CONTROL.002*, and so on.

If you copy files to a fixed disk, BACKUP automatically creates a directory called \BACKUP, which contains the BACKUP and CONTROL files. This directory is placed in the root of the target fixed disk so you must have Create and Write access to the root of the target drive.

The BACKUP utility automatically formats an unformatted floppy disk before saving backup copies on it.



## Making Backup Copies of Subdirectories

If you use the BACKUP utility without options, only the files or directory you specify are copied. To copy several directories or an entire disk, you can use the /S option. The /S option enables you to create backup copies of subdirectories. For example, to back up your fixed disk (drive C), including all subdirectories, to floppy disks in drive A, type the following:

```
BACKUP C: A: /S
```

The original directory structure is maintained while all files and directories are copied to drive A.

## Making Backup Copies of Files Modified Since the Last Backup Operation

The BACKUP utility can use the archive file flag to determine which files to back up. If you use the /M option, only those files that have been modified since the last backup operation are copied to the backup disk. However, the previous backup files will be deleted from the backup disk (unless you use the /A option, as described later).

Whenever you use BACKUP, those files for which the archive flag is set are found and copied. The BACKUP utility then turns off the archive flag for those files. When you make a backup copy of the same set of files and use the /M option, the archive flag is set only for those files that have been created or modified since the last backup operation. Therefore, only they will be backed up. (The /M option may not perform as described here if you use the ATTRIB utility to modify the archive flags of the files.)



You can also use the /M option to make copies of directories. When you use this option, the BACKUP utility makes a copy of a directory only if its extended attributes have changed since the last backup operation.

## **Adding Files to the Backup Disk**

By default, the contents of the backup floppy disk or backup directory are erased before the backup files are added to it. If you want to add files to an existing backup disk or directory without modifying the current contents, use the /A option. Suppose your backup floppy disk contains the contents of the ENG directory and you need to make a backup copy of the MKT directory. You could make a second backup disk or you could add the contents of MKT to the existing backup disk by using the /A option. To add the second directory to the backup disk, type the following:

```
BACKUP C:\MKT A: /A
```

The *BACKUP* and *CONTROL* files now contain backup copies of the contents of both the ENG and MKT directories. Any other files that were on the backup disk remain unchanged.

## **Making Backup Copies of Files Modified After a Certain Date and Time**

You may want to back up only those files that were modified after a certain date and time. To do this, use the /D:*date* option and, optionally, the /T:*time* option. Note that /T cannot be used without /D.

Suppose you wanted to back up only those files that were modified after the beginning of 1989. You would run the BACKUP utility and specify the date and time as follows:

```
BACKUP C: A: /D:12-31-88 /T:23:59
```

Any files that were modified after 11:59 P.M. on December 31, 1988 would be copied.

## Making a Backup Log

The `/L[:logfile]` option lets you create a log of the files you have copied by using the BACKUP utility. This log contains a list of entries, one entry per file. Each entry contains the number of the backup disk and the name of the file. This information can be used to restore a particular file from a floppy disk.

Unless you specify otherwise, the log file is created in the root directory of the source disk or source drive. If you do not specify a log file name, the log file is named `\BACKUP.LOG`. For example, suppose you want to back up the contents of the MKT directory on drive C to a floppy disk in drive A. To create a backup log on drive C called `MKTLOG.001`, type the following:

```
BACKUP C:\MKT A: /L:MKTLOG.001
```

## Preserving Extended Attributes

In MS OS/2 *MULTIUSER*, some applications attach additional information, such as author, application type, and file history, to your files and directories. This type of information is called extended attributes and is used by other applications, the file system, or the operating system itself. The BACKUP utility automatically preserves the extended attributes of a file or directory.

## Preserving Security Attributes

The MS OS/2 *MULTIUSER* security attributes are automatically preserved using the BACKUP utility. These attributes are not saved if you backup using COPY or XCOPY, and may not be saved with other backup mechanisms.

## Restoring Backup Files

To restore backup files to a disk, use the RESTORE utility. The RESTORE utility can restore files between similar or different disk types. Many of the options of the RESTORE utility are analogous to the options of the BACKUP utility.

## Restoring Files to a Disk

To restore files that were copied by using the BACKUP utility, type RESTORE followed by the drive that contains the disk with the backup files (the source drive) and the name of the drive or directory that these files will be copied to (the destination drive). You can restore files only to the directories from which they were copied.

For example, if you have a floppy disk that contains backup files, you need only insert it in drive A and type the following to restore the files to the BAK directory on drive C:

```
RESTORE A: C:\BAK
```

## Restoring Subdirectories

If you make backup copies of subdirectories by using the /S option of the BACKUP utility, you can restore these subdirectories by using the /S option of the RESTORE utility. For example, to restore all of the backup files in drive A to the current directory in drive C, type the following:

```
RESTORE A: C: /S
```

## Prompting the User Before Restoring Files

The directory being restored may contain read-only files or files that have been modified since the last backup operation; you may not want to restore these files. The /P option displays a warning message and prompts you with "Replace the file (Y/N)?" To restore the backup file, type Y for yes. To leave the file unchanged, type N for no.

## Selecting Files to Restore

Just as you can make backup copies based on the time or date a file was created or last modified, you can also restore files based on the time or date. The options that you can use with RESTORE are as follows:

*/B:date*

Restores only those files that were last modified on or before date.

*/A:date*

Restores only those files that were last modified on or after date.

*/E:time*

Restores only those files that were last modified at or earlier than time.

*/L:time*

Restores only those files that were last modified at or later than time.

*/M*

Restores only those files that were modified after the last backup.

You can combine options, if you like.



These options work only if the *filenames* on the *source* drive are the same as those on the *destination* drive. In other words, if you are restoring files to an empty directory or to a directory that contains *filenames* that are different from the backup *filenames*, these options do not apply. For example, suppose that you want to update a phone list periodically from a backup floppy disk that is made from the company's master phone lists. Suppose also that you want to restore the list to a directory named PHONE on drive C. To update those phone lists that have not been modified since a specified date, you could use the /B and /M options, as follows:

```
RESTORE A: C: /B:5-14-89 /M
```

The /B option specifies only those files that were last modified on or before May 14, 1989. The /M option specifies those files that have the archive flag set, which means that the file has been modified since the last backup operation.

## Restoring Files That Are Not on the Destination Drive

To restore only those files from the backup disk that do not currently exist on the *destination* drive, use the /N option.

For example, suppose you have the files *TEMP1.BAK*, *TEMP2.BAK*, and *TEMP3.BAK* on a backup disk and that *TEMP1.BAK* and *TEMP2.BAK* already exist in the *destination* directory. If you use the /N option, only *TEMP3.BAK* will be restored.

## Restoring Extended Attributes and Security Attributes

The RESTORE utility, like the BACKUP utility, preserves the extended attributes and security attributes of a file or directory. If you are restoring files that have extended or

security attributes to a file system that does not support them, you may want to use the /F option with the RESTORE utility. For more information about the RESTORE utility, see the *Citrix MULTIUSER Command Reference*.

## WORKING WITH FIXED AND FLOPPY DISKS

MS OS/2 *MULTIUSER* provides the following utilities to help you work with disks:

<u>Utility</u>	<u>Purpose</u>
FORMAT	Formats a disk.
DISKCOPY	Copies the contents of a floppy disk to another floppy disk.
LABEL	Assigns a volume label to a disk.
DISKCOMP	Compares the contents of two floppy disks.

These utilities are described in the following sections.

### Formatting a Disk

Before you can use a fixed or floppy disk, you must prepare it by using the FORMAT utility. You can use the FORMAT utility on new disks and on disks that have been formatted before. When you format a disk, any information it contained is lost. MS OS/2 *MULTIUSER* formats a fixed disk to use HPFS when you use the /FS option with the FORMAT utility. You cannot format a floppy disk to use HPFS. If you are running HPFS and you use the FORMAT utility to format a floppy

disk, the disk is formatted as a FAT floppy disk. Suppose you want to format logical drive D on your fixed disk to use HPFS. To do this, type the following:

```
FORMAT D: /FS:HPFS
```

There is no need to use the */T:tracks* or */N:sectors* options. HPFS always formats the disk to use the maximum number of tracks and sectors.

## Formatting a Floppy Disk

Before you can use a floppy disk, you must prepare it for use by using the FORMAT utility. You can also format a previously formatted disk. Formatting a disk erases all of its contents.

Before you format your disk, make sure that you know what type of disk you have and what type of disk drive is in your computer. The following list shows the types of floppy disks that can be formatted with the FORMAT utility:

<u>Disk Size</u>	<u>Type</u>
5¼-inch	360-kilobyte (low-density)
5¼-inch	1.2-megabyte (high density)
3½-inch	720-kilobyte (low-density)
3½-inch	1.44-megabyte (high-density)

With the default setting, the FORMAT utility will format a disk according to the size and type of drive in your computer. If you do not specify options and if your computer has a low-density drive, the FORMAT utility functions as though you are formatting a low-density disk; if your

computer has a high-density drive, FORMAT functions as though you are formatting a high-density disk. (If your computer has more than one type of drive, the density level that FORMAT sets itself to process will depend on which drive you tell FORMAT to use.) Later in this chapter you'll learn about options that override the default setting.

## Formatting a 5¼-Inch Disk

To format a 5¼-inch disk (low- or high-density) in a disk drive of the same size, type FORMAT followed by the letter of the drive that contains the disk. If you want to format a disk in drive A, type the following:

```
FORMAT A:
```

A message will then appear asking you to insert a floppy disk in drive A and press **ENTER**, at which point formatting will begin. Usually disks are given a volume label that identifies the disk. This label can be up to 11 characters in length. After a disk is formatted, the FORMAT utility automatically prompts you to type the volume label. Then another message will appear, asking whether you want to format another disk. If you prefer, you can specify the volume label as an argument to format by using the option */V:label* after the drive letter. For example, to give the label REPORTS to the disk in drive A, type the following:

```
FORMAT A: /V:REPORTS
```

Because you have already specified the volume label, you will not be prompted to type it after the disk is formatted. Although a high-density (1.2-megabyte) drive is designed to format high-density disks, you can also format a 5¼-inch



double-sided low-density (360K) disk by using the option /4. For example, to format a low-density disk in drive B, which is a high-density drive, type the following:

```
FORMAT B: /4
```

**NOTE:** This option is designed to format low-density disks that will be used in high-density drives. If you format a disk with the /4 option in a high-density (1.2-megabyte) drive, you might not be able to read that disk in a low-density (360K) drive.

## Formatting a 3½-Inch Disk

The FORMAT utility has two options that are used to format 3½-inch disks: /T:*tracks* and /N:*sectors*. These options are normally used to format disks that are of a lower density than the drive (for example, to format a 720K disk in a 1.44-megabyte drive).

The /T:*tracks* option formats a disk to the number of tracks specified. Tracks is the number of tracks per disk. For 720K and 1.44-megabyte disks, this value is 80. If you do not specify this option, FORMAT uses the size of the drive to determine how many tracks the disk should have.

The /N:*sectors* option formats a disk to the number of sectors specified. Sectors is the number of sectors per track. For 720K disks, this value is 9 and for 1.44-megabyte disks, this value is 18. For example, to format a 3½-inch low-density (720K) disk in drive A, which is a high-density drive, type the following:

```
FORMAT A: /N:9
```

You cannot do the opposite, however. You cannot format a high-density disk in a low-density drive.

## Copying a Floppy Disk

To copy the contents of a floppy disk in one drive (the *source* drive) to a floppy disk in another drive (the *destination* drive), you can use the DISKCOPY utility. If the *destination* disk is unformatted, DISKCOPY formats the disk with the same number of sides and sectors per track as the source disk. Use this utility with care, because DISKCOPY destroys the existing contents of the *destination* disk. To run the DISKCOPY utility, type DISKCOPY followed by the names of the *source* and *destination* drives. The *source* and *destination* drives can be the same; if you do not specify any drives, DISKCOPY uses the current drive as both *source* and *destination*. The DISKCOPY utility prompts you to insert the *source* and *destination* disks at appropriate times and waits for you to press a key before continuing. If errors are found on either disk, a message appears on the screen, describing the drive, track, and side where the error was found. After this, the copying operation continues. When it finishes copying the first disk, DISKCOPY asks you whether you want to copy another disk.

The DISKCOPY utility is most often used to make backup copies of floppy disks. DISKCOPY copies an entire floppy disk faster than the COPY command does. For example, many applications require you to make backup copies of the installation disk. To do this using DISKCOPY, insert the installation disk in drive A and a blank formatted disk in drive B. Then type the following:

DISKCOPY A: B:

Or, if your computer has only one floppy-disk drive, specify drive A as both the *source* and the *destination* drives:

DISKCOPY A: A:

The DISKCOPY utility prompts you to insert the *source* or *destination* disk at appropriate times. When DISKCOPY has finished copying the disk, a message appears asking whether you want to copy another disk.

The copying is done track by track and produces an identical copy of the original disk. DISKCOPY does not work on fixed disks. The *source* and *destination* disks cannot be virtual disks. Certain restrictions apply concerning which types of disks can be copied in certain disk drives. For more information about types of disks and disk drives, see "Formatting a Floppy Disk" earlier in this chapter.

## Displaying the Volume Identification

Each disk has a volume label, which is a name you assign to the disk during formatting. You might want to give a floppy disk a unique label to help you identify the disk later. Also, if you have set up logical drives, the volume label assigned to each of those drives can describe the contents of the drive (for example, drive D might be called SMITH and drive E be called JONES).

If a disk has been formatted using MS OS/2, version 1.21, or MS OS/2 *MULTIUSER*, it also has a volume serial number. This number is a unique number assigned by MS OS/2 *MULTIUSER*, and it can be changed only by reformatting the disk. To view the volume identification for your current disk, type the following:

VOL

You can also view the volume identification for a different disk, such as the one on drive C, by typing the drive letter after the VOL command:

VOL C:

You can view the volume identification for more than one drive at a time by typing more than one drive letter after the VOL command:

VOL D: C:

If the VOL command cannot find one of the specified drives, it displays an error message telling you which drive could not be found and continues to display information for the other drive.

You can set the volume label by using the LABEL utility, which is described in the next section.

## Assigning or Changing the Volume Label

To assign a volume label to a floppy or fixed disk or to change the existing label, use the LABEL utility. To use this utility, type LABEL followed by the drive letter and the label to be assigned to the disk. If you omit the drive letter, the label will be assigned to the current drive. For example, suppose that you want to change the volume label of drive D from APPS to MKTG. Type the following:

LABEL D:

The current label is displayed and you are prompted for the new volume label:

The volume label in drive D is APPS. Enter a volume label of up to 11 characters or press **ENTER** for no volume label update.



Type MKTG and press ENTER. If you now type VOL D: to view the label, you will see that the label has been changed to MKTG. You can type the volume label on the command line as follows:

LABEL D: MKTG

No messages are displayed. The prompt appears again after the operation is complete.

## Comparing Floppy Disks

To compare the contents of a floppy disk in the *source* drive with the contents of a floppy disk in the *destination* drive, use the DISKCOMP utility. The comparison is done track by track and DISKCOMP automatically determines the number of sides and sectors per track based on the format of the *source* disk. The *source* and *destination* disks must be of the same type. DISKCOMP does not work on fixed disks. For example, suppose that you receive an update package for a word processing program. When you are in the middle of making a backup copy of the updated version, you are interrupted, and the floppy disk you have just copied gets mixed in with disks that contain old versions of the program. To determine which disk you just copied, place the update disk in drive A and one of the unknown disks in drive B and type the following:

DISKCOMP A: B:

If your computer has only one floppy disk drive, specify drive A as both the *source* and *destination* drives:

DISKCOMP A: A:

The DISKCOMP utility prompts you to insert the *source* and *destination* disks at appropriate times during the disk comparison. If it finds a place where the contents of the two disks are not identical, an error message appears, telling you the side and track that are different.

If the disks are identical, a message appears at the end of the comparison, telling you that the disks are identical. After DISKCOMP is through running, a message appears, asking you whether you want to compare two more disks.

## QUERYING SYSTEM STATUS

### Introduction

The QUERY utilities are used to display current information about the system such as active users or the current allocation of resources. Information in the various profiles is not displayed by these utilities. Profile information can be queried using the /Q option on the CONFIG command line. The QUERY utilities do not affect the state of the system. They simply present the information in an organized fashion.

The utilities are described in the following sections. Refer to the QUERY commands in the *Citrix MULTIUSER Command Reference* for detailed information on these utilities and their options.

The command "QUERY" can invoke any one of the query utilities. The command line format is:

```
QUERY [AUDIT | EVENTS | HOST | LIMITS | LOGIN |  
MEMORY | PRINT | PROCESS | SESSION | TERMINAL | USER]  
[/?]
```

## QUERY AUDIT

QUERY AUDIT examines the system-wide security audit events that are currently being logged. This only queries the system-wide auditing flags. Audit flags can also be placed on users and resources using CONFIG USER and CONFIG ACCESS. These commands, CONFIG USER and CONFIG ACCESS, must be used to query the user and resource audit flags.

Auditing can be turned on and off using CHANGE EVENTS. If auditing is off, the audit flags displayed here will not be used until audit event logging is turned on.

The QUERY AUDIT command has no parameters or options.

For more information on the QUERY AUDIT command, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY EVENTS

QUERY EVENTS displays the current state of event logging. The events being logged can be changed permanently using CONFIG SYSTEM. Event logging can also be controlled using CHANGE EVENTS.

The QUERY EVENTS command has no parameters or options.

For more information on the QUERY EVENTS command, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY HOST

QUERY HOST displays information about the host that the current user is logged into. This information includes:

- The hostname
- The total number of active terminals
- The total number of active logins
- The total number of sessions
- The total number of processes
- The startup time of the system.

See the figure below for a sample output of QUERY HOST.

HOSTNAME	TERMINALS	LOGINS	SESSIONS	PROCESSES	BOOT TIME
System1	9	9	21	28	06/20/91 06:20

HOSTNAME identifies the system. TERMINALS specifies the number of active terminals on the system. LOGINS specifies the total number of logins in the system. SESSIONS specifies the total number of sessions in the system. PROCESSES specifies the total number of processes in the system. BOOT TIME indicates the date and time the system was started.



An example of the QUERY HOST command is:

**QUERY HOST**

Displays information about the current host.

## **QUERY LIMITS**

QUERY LIMITS displays the user resource limits and current totals for one or more users in the system. The minimum and maximum limit and current total for each resource are displayed.

Some examples of the QUERY LIMITS command follow:

**QUERY LIMITS \*.\***

Displays user resource limits and current totals for all users in the system.

**QUERY LIMITS DAVIDH.WRKGRP**

Displays user resource limits and current totals for user DAVIDH in group WRKGRP. An example of this command follows:

```
[C:\USR\CHRISL]query limits *.wrkgrp
**** CHRISL.WRKGRP ****
```

RESOURCE	MINIMUM	LIMIT	MAXIMUM	LIMIT	CURRENT	TOTAL
Virtual Memory (Kb)		1024		4096		156
Semaphores		5		64		4
Threads		20		128		3
Sessions		0		10		1
Logins		0		1		1
File Handles		30		128		2

```
**** MIKEH.WRKGRP ****
```

RESOURCE	MINIMUM	LIMIT	MAXIMUM	LIMIT	CURRENT	TOTAL
Virtual Memory (Kb)		1024		4096		239
Semaphores		5		64		5
Threads		20		128		5
Sessions		0		10		1
Logins		0		1		1
File Handles		30		128		6

### QUERY LIMITS \*.WRKGRP

Displays user resource limits and current totals for all users in the group WRKGRP.

QUERY LIMITS is useful when trying to decide what values are needed for user resource limits. The Administrator can call up a user's standard set of applications and then check resource usage via this command. The current total value plus a buffer is used as a base for the user's minimum resource limit.

For more information on the QUERY LIMITS command, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY LOGIN

QUERY LOGIN displays such information about a login as:

- The *loginname* that owns the login.
- The *terminalname* where the login is connected.
- The *loginID* of the login.
- The terminal where the login was previously connected, if different from the current terminal.
- The time when the login was disconnected, if the login has been disconnected.

Sample output for QUERY LOGIN follows:

LOGINNAME	TERMNAME	LOGINID	PREV TERM	DISCONNECT	TIME
>annm.wrkgrp	serial2	15			
miked.wrkgrp	disc	9	serial1	01/29/91	17:35
admin.wrkgrp	console	7			
system.system	disc	1	disc	01/29/91	6:51
system.system	serial1	16			

**NOTE:** The > character indicates the current login. LOGINNAME identifies the *username* and *groupname* of each login listed. TERMNAME identifies the *terminalname* where the login is connected. LOGINID is the ID of the

login. PREV TERM specifies the *terminalname* at which the login was previously connected. DISCONNECT TIME specifies the time at which the login was disconnected.

Some examples of the QUERY LOGIN follow:

#### QUERY LOGIN

Displays information about all logins belonging to the current *loginname*.

#### QUERY LOGIN 5

Displays information about *loginID* 5.

For more information on the QUERY LOGIN command, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY MEMORY

QUERY MEMORY displays information about the allocation of memory in the system. This includes the amount of system memory being used and memory that is currently swapped out onto disk. Total free physical memory can also be displayed. If no *loginname* is specified on the command line, the current *loginname* is used.

Memory usage for specific applications can be examined using the QUERY MEMORY command. This command summarizes applications and dynamic link libraries' memory usage for users in the system. The information is broken down into the amount of memory present in physical memory and the amount that is swapped to external memory (disk). Because MS OS/2 *MULTIUSER* is a shared logic system, the information is displayed in a private memory group and shared memory group:



### Private Memory

This is the private data that application and dynamic link libraries allocate. User resource limits are based on this type of memory. The information is displayed on a per user basis.

**NOTE:** Because the QUERY MEMORY utility uses different methods to obtain its values than the QUERY LIMITS utility, these numbers will not match.

### Shared Memory

Application code and shared data are grouped together under the shared memory classification. These types of memory are allocated only once in physical memory and are shared among all users of the application/dynamic link library.

The /FREE option can be useful in determining how much free physical memory is available.

The following figure details the outputs of the QUERY MEMORY command.

Application Private Memory:						
LOGINNAME	MODULE	PID	SEGMENTS	RESIDENT	SWAPPED	TOTAL
anng.wrkgrp	psel.exe	12	39	162449	9402	171851
anng.wrkgrp	cmd.exe	14	30	58949	9621	68570
gregg.citrix	psel.exe	10	39	163171	9402	172573
gregg.citrix	cmd.exe	13	29	58965	9621	68586
gregg.citrix	qmemory.exe	22658	24	49647	9737	59384
system.system	mumproc.exe	5	53	99952	469	100421
system.system	hed.exe	6	27	40056	9402	49458
system.system	cmd.exe	11	29	53053	9621	62674
Total:			270	686242	67275	753517
Application Shared Memory:						
TYPE	MODULE		SEGMENTS	RESIDENT	SWAPPED	TOTAL
(Shared Code)	psel.exe		3	73216	0	73216
(Shared Code)	mumproc.exe		1	47698	0	47698
(Shared Code)	cmd.exe		5	78558	0	78558
(Shared Code)	hed.exe		2	13322	0	13322
(Shared Code)	qmemory.exe		1	16432	0	16432
Total:			12	229226	0	229226

Some examples of the QUERY MEMORY command follow:

#### QUERY MEMORY \*.\* /DLL

Displays all users' memory usage by application.  
Dynamic link library memory usage is also displayed.

#### QUERY MEMORY GREGG.WRKGRP

Displays application memory usage for user GREGG in group WRKGRP.

#### QUERY MEMORY \*.WRKGRP

Displays application memory usage for all users in the group WRKGRP.

For more information on the QUERY MEMORY utility, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY PRINT

QUERY PRINT displays the current print queues in the system and the print jobs in each queue for the *loginname*, if there are any. If no *loginname* is specified, all print jobs are displayed.

The default for QUERY PRINT is to display the print jobs for the entire system.

The *loginname* can contain wildcard characters.

Sample output for QUERY PRINT follows:

Name	Job ID	Size	Status
<hr/>			
LPT1Q	3 Job(s)		Queue Active
└ SCOTTK.TPSS	1	42	Printing
└ EDJ.TPSS	2	3303	Queued
└ ANNM.MUM	3	13003	Queued
LPT2Q	0 Job(s)		Queue Active
LPT3Q	2 Job(s)		Queue Held
└ JOELS.TPSS	5	5363	Queued
└ GEORGEV.BETA	6	455	Queued

## QUERY PROCESS

QUERY PROCESS displays such information about a process as:

- The user context in which the process is running, including the *loginname*, *loginID*, and *sessionID*.
- The *terminalname* where the process is running.
- The process state.

- The *processID*.
- The parent *processID*.
- The process *filename*.

Sample output for QUERY PROCESS follows:

LOGINNAME	TERMNAME	LID	SID	STATE	PPID	PID	
annm.wrkgrp	serial2	15	1	block	4	35	psel.exe
> annm.wrkgrp	serial2	15	2	ready	4	36	cmd.exe
miked.wrkgrp	disc	9	1	block	4	18	psel.exe
miked.wrkgrp	disc	9	2	block	4	27	cmd.exe
miked.wrkgrp	disc	9	2	exit	27	52	r2eargs.exe
miked.wrkgrp	disc	9	3	ready	4	51	little.exe
miked.wrkgrp	disc	9	4	block	4	54	cmd.exe
admin.wrkgrp	console	7	1	block	4	15	psel.exe
admin.wrkgrp	console	7	2	block	4	37	cmd.exe
admin.wrkgrp	console	7	3	block	4	39	cmd.exe
admin.wrkgrp	console	7	3	ready	39	40	sh.exe
system.system	disc	1	1	block	0	4	mumproc.exe
system.system	disc	1	2	block	4	5	hed.exe
system.system	serial1	16	1	block	4	56	login.exe

**NOTE:** The > character indicates the current process after QUERY PROCESS completes. LOGINNAME identifies the *username* and *groupname* of each user listed. TERMNAME identifies the *terminalname* where the user is logged into the system. LID and SID correspond to the *loginID* and *sessionID*, respectively. STATE indicates the state of the process. The states of a process include the following:



blocked (BLOCK), ready (READY), terminating (EXIT), or CPU-starved (STARVE). PPID identifies the parent *processID*. PID identifies the *processID* of the program specified by the following *filename*.

Some examples of the QUERY PROCESS command follow:

#### QUERY PROCESS

Displays information about all processes running in logins belonging to the current *loginname*.

#### QUERY PROCESS 17

Displays information about *processID* 17.

#### QUERY PROCESS \*

Displays information about all processes in the system.

For more information on the QUERY PROCESS utility, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY SESSION

QUERY SESSION displays such information about a session as:

- The user context of the session, including the *loginname* and *loginID*.
- The *terminalname* where the session is located.
- The *sessionID*.
- The session state.
- The session title.

Sample output for QUERY SESSION follows:

LOGINNAME	TERMNAME	LID	SID	STATE	TITLE
annm.wrkgrp	serial2	15	1	bg	program selector
> annm.wrkgrp	serial2	15	2	fg	qsession.exe
miked.wrkgrp	disc	9	1	bg	program selector
miked.wrkgrp	disc	9	2	bg	r2eargs.exe
miked.wrkgrp	disc	9	3	det	little.exe
miked.wrkgrp	disc	9	4	fg	cmd.exe
admin.wrkgrp	console	7	1	bg	program selector
admin.wrkgrp	console	7	2	bg	* os/2 command prompt *
admin.wrkgrp	console	7	3	fg	sh.exe
system.system	disc	1	1	fg	multi-user manager
system.system	disc	1	2	det	hed.exe
system.system	serial1	16	1	fg	login.exe

**NOTE:** The > character indicates the current session. LOGINNAME identifies the *username* and *groupname* of each session listed. TERMNAME identifies the *terminalname* where the login is connected. LID and SID correspond to the *loginID* and *sessionID*, respectively. STATE specifies whether the session is foreground (fg), background (bg), or detached (det). TITLE is the title assigned to the session, which defaults to the program *filename* if no title was provided when the session was created.

Some examples of the QUERY SESSION command follow:

#### QUERY SESSION

Displays information about all sessions in all logins belonging to the current *loginname*.

**QUERY SESSION 1**

Displays information about all sessions with *sessionID* 1.

**QUERY SESSION \***

Displays information about all sessions in the system.

For more information on the QUERY SESSION utility, refer to the *Citrix MULTIUSER Command Reference*.

**QUERY TERMINAL**

QUERY TERMINAL displays information about a specific *terminalname* if that *terminalname* is active.

QUERY TERMINAL displays such information about the terminal as:

- The *terminalname*.
- The terminal type.
- The device name.
- Comments associated with the terminal.

Sample output for QUERY TERMINAL follows:

TERMINALNAME	TYPE	DEVICE	
serial3	tvi965	com3	mike's office
serial1	ibm3151	com2	brad's office
> serial2	wyse60	com1	ann's office
console	console	con	administrator's station

**NOTE:** The > character indicates the current terminal. TERMINALNAME specifies the name that is assigned to the terminal. TYPE indicates the terminal type as identified in the terminal profile. DEVICE is the device name that the terminal is assigned. The comment following the terminal information is from the terminal profile.

Some examples of the QUERY TERMINAL command follow:

#### QUERY TERMINAL

Displays information about all active terminals.

#### QUERY TERMINAL SERIAL1

Displays information about terminal SERIAL1.

For more information on the QUERY TERMINAL utility, refer to the *Citrix MULTIUSER Command Reference*.

## QUERY USER

QUERY USER displays such information about a user who is logged into the system as:

- The *loginname* and *loginID*.
- The *terminalname* where the login is connected.



- The time since the last keystroke, in minutes.
- The time when the user logged into the system.

Sample output for QUERY USER follows:

LOGINNAME	TERMNAME	LOGINID	IDLE TIME	LOGIN TIME
bradp.wrkgrp	serial1	11	2	01/29/91 8:01
> annm.wrkgrp	serial2	15	.	01/29/91 7:30
miked.wrkgrp	disc	9	1	01/29/91 7:25
admin.wrkgrp	console	7	4	01/29/91 13:57

**NOTE:** The > character indicates the current user.

LOGINNAME identifies the *username* and *groupname* of each user listed. TERMNAME identifies the *terminalname* where the user is logged into the system. LOGINID identifies the user login. IDLE TIME is a measure of interaction between the user and the system by indicating the time in minutes since the last keystroke. A dot (.) indicates that the time is less than a minute. LOGIN TIME is the time when the user logged into the system.

Some examples of the QUERY USER command follow:

#### QUERY USER

Displays information about all users who are logged into the system.

#### QUERY USER DAVIDH

Displays information about *username* DAVIDH.

For more information on the QUERY USER command, refer to the *Citrix MULTIUSER Command Reference*.

## CHANGING SETTINGS

### Introduction

The CHANGE utilities are used to change current settings. Changes made using the CHANGE utilities are temporary. If the changes made are for a user login (LIMITS), the changes last until the user logs out. All changes are lost when the system restarts or the CONFIG or RESET utilities are used to modify the object that was changed with the CHANGE utility. Information in the profiles is not modified by the CHANGE utilities.

The CHANGE utilities are described in the following sections. Refer to the CHANGE commands in the *Citrix MULTIUSER Command Reference* for detailed information on these utilities and their options.

The command CHANGE can invoke any one of the CHANGE utilities. The command line format is:

```
CHANGE [AUDIT|EVENTS|LIMITS|PRNMODE|TERMINAL] [/?]
```

### Audit

CHANGE AUDIT will temporarily change the system-wide security audit events that are being logged.

CONFIG SYSTEM establishes audited events and places them in the system profile. When the system is started, the current audit logging state is taken from the system profile. The selection of audit events being logged can be changed any time during system operation using CHANGE AUDIT. This change will only be in effect until the system is restarted.

This only changes the system-wide auditing flags. Audit flags can also be placed on users and resources using CONFIG USER and CONFIG ACCESS. These commands, CONFIG USER and CONFIG ACCESS, must be used to change the user and resource audit flags.

Auditing can be turned on and off using CHANGE EVENTS. If auditing is off, the audit flags established here will not be used until audit event logging is turned on.

Use this command with caution. Auditing system-wide granted access of many access types can generate a great amount of logging activity and can severely impact system performance. It is far more common to audit only denied accesses.

## **Events**

CHANGE EVENTS temporarily changes which events are being logged.

This sets or clears events that are being logged, but only until the next reboot. To permanently change the event log selection, see CONFIG SYSTEM.

The events specified are added (/SET) or removed (/CLR) from the already existing list. At least one parameter, /SET or /CLR must be specified.

## **User Limits**

User resource limits can be temporarily changed using the CHANGE LIMITS command. This command changes the user resource limits for one or more users in the system. The minimum or maximum limit for any resource can be

changed. Changes in limits will not be written to the user's profile file; therefore, the new limits will remain only until logins under this *loginname* are terminated. To change a user's limits permanently, use the CONFIG USER command.

Some examples of the CHANGE LIMITS command follow:

CHANGE LIMITS /MAXTHREADS:32 \*.\*

Changes the maximum number of threads to 32 for all users in the system.

CHANGE LIMITS /MINMEMORY:1024 DAVIDH.WRKGRP

Changes the minimum guaranteed limit for virtual memory to 1024 KB (1 MB) for user DAVIDH in group WRKGRP.

CHANGE LIMITS /MAXFILES:50 \*.WRKGRP

Changes the maximum number of open files to 50 for all users in the group WRKGRP.

The CHANGE LIMITS command is useful when a user receives "out of resource" hard error popups. The Operator or Administrator can temporarily increase the user's limits using the CHANGE LIMITS command and the application should then continue normally.

## Terminal

The CHANGE TERMINAL command is used to change the terminal parameters of the terminal for the duration of the login. It does not change the permanent configuration. To change the permanent configuration, execute the CONFIG TERMINAL command.



Changing the terminal type is provided primarily to allow for dial-in terminals. A terminal may be configured as a generic dial-in terminal, which means that the terminal driver uses a subset of the available functions that are provided by most terminals. After login, a user may use the CHANGE TERMINAL utility to install the driver that matches the specific terminal at which the user is working. This allows the user to take advantage of the specific capabilities of the terminal. The terminal types are listed in Appendix D, "Setting Up Specific Terminal Types."

## RESETTING SETTINGS

### Introduction

The RESET utilities are used to undo any changes made by the CHANGE utilities. Information in the profiles is not modified by the RESET utilities.

The RESET utilities are described in the following sections. Refer to the RESET commands in the *Citrix MULTIUSER Command Reference* for detailed information on these utilities and their options.

The command RESET can invoke any of the RESET utilities. The command line format is:

```
RESET [AUDIT|EVENTS|LIMITS|TERMINAL] [/?]
```

### Audit

RESET AUDIT resets the system-wide security audit events that are currently being logged to be equal to those specified in the system profile.

CONFIG SYSTEM establishes the audited events and places them in the system profile. When the system is started, the current audit logging state is taken from the system profile. The selection of audit events being logged can be changed any time during system operation using CHANGE AUDIT. The RESET AUDIT command resets the state of the auditing to that specified in the system profile.

This only resets the system-wide auditing flags. Audit flags can also be placed on users and resources using CONFIG USER and CONFIG ACCESS. This command does not affect the user and resource audit flags.

Auditing can be turned on and off using CHANGE EVENTS. If auditing is off, the audit flags established here will not be used until audit event logging is turned on.

## Events

RESET EVENTS resets the events that are currently being logged to be equal to those specified in the system profile.

CONFIG SYSTEM establishes the logged events and places them in the system profile. When the system is started, the current event logging state is taken from the system profile. The selection of events being logged can be changed any time during system operation using CHANGE EVENTS. The RESET EVENTS command resets the state of the event logging to that specified in the system profile.

## User Limits

User resource limits can be reset to their initial values, found in the user's profile file, using the RESET LIMITS command. This command resets the user resource limits for one or more users in the system. The minimum or maximum limit for any

resource can be reset. This command is useful if a user's limits were temporarily changed with the CHANGE LIMITS command and you wish to reset the limits to their initial login values while the user remains logged in.

Some examples of the RESET LIMITS command follow:

RESET LIMITS /MAXTHREADS \*.\*

Resets the maximum number of threads to its initial login value for all users in the system.

RESET LIMITS /MINMEMORY DAVIDH.WRKGRP

Resets the minimum guaranteed limit for virtual memory to its initial login value for user DAVIDH in group WRKGRP.

RESET LIMITS /MAXFILES \*.WRKGRP

Resets the maximum number of open files to its initial login value for all users in the group WRKGRP.

## Terminal

The RESET TERMINAL function is used to reset a terminal to a known, initial state. This utility is used when the terminal is malfunctioning or the terminal appears to be down. The terminal is reset, the underlying hardware is reset, and the terminal drivers and terminal capability data are reloaded and reinitialized.

The RESET TERMINAL function is also used to take the terminal out of the disabled initial state. There is a configuration parameter that allows for the terminal to be configured to start in the disabled state. Again, although the terminal is started by the RESET TERMINAL utility when the terminal is in the disabled initial state, the permanent configuration record is not changed by executing the RESET TERMINAL function. CONFIG TERMINAL is used for this purpose.

Finally, the RESET TERMINAL utility is used to allow a terminal to be configured without having the terminal or the terminal cables available. An attempt is made during the running of CONFIG TERMINAL to bring the terminal up. If this fails, CONFIG TERMINAL proceeds as normal. When the terminal is finally physically connected, the RESET TERMINAL utility must be run.

## TERMINATING A PROCESS

Use the KILL utility to terminate a process. KILL can be used to terminate all processes in a login or at a terminal. KILL forces the specified process to end, so the process may lose data. The KILL utility should only be used if there is no alternative but to abort the process.

Some examples of the KILL command follow:

KILL 14

Terminates the process with *processID* 14.

KILL \ZEUS.WRKGRP\7

Terminates all processes in *loginID* 7 belonging to the *loginname* ZEUS.WRKGRP.

For more information on the KILL utility, refer to the *Citrix MULTIUSER Command Reference*.

## SENDING MESSAGES TO USERS

The message utility, MSG, lets you send notes to other users who are currently logged in. To determine if the user is currently logged in, use the QUERY USER utility, which displays a list of *loginnames* that are currently logged in. Then when you use MSG, you identify the login(s) that you wish to receive your message.



You may use MSG to broadcast a message by specifying a wildcard for the receiving logins.

Use MSG at the CMD command prompt. In the example, the message "Let's meet as soon as possible to discuss those work items." is sent to the *loginname* ZEUS.WRKGRP.

```
MSG ZEUS.WRKGRP Let's meet as soon as possible to
discuss those work items.
```

Both the message and the loginname of the sender is displayed in the login of the receiver.

For more information on the MSG utility, refer to the *Citrix MULTIUSER Command Reference*.

## SETTING DATE AND TIME

### Setting and Displaying the Date

You can set and display the system date by using the DATE command. MS OS/2 *MULTIUSER* uses this date to update the directory list whenever you create or change a file or directory. The date that you set applies to all sessions. To use the DATE command, type the following:

```
DATE
```

You'll see a message similar to this:

```
The current date is Fri 3-24-1989
Enter the new date: (mm-dd-yy) _
```

To keep the current date, press **ENTER**. To change the date, type the new date at the cursor, separating the month, day, and year with hyphens (-), slashes (/), or periods (.). For example, the dates 6-1-89, 6/1/89, and 6.1.89 are equivalent.

You can also specify the date directly after the **DATE** command, as follows:

```
DATE 3-1-89
```

## Setting and Displaying the Time

You can set and display the system time with the **TIME** command. This command sets the internal clock in your computer. MS OS/2 *MULTIUSER* uses this time to update the directory list whenever you create or change a file or directory. The time that you set applies to all sessions. To use the **TIME** command, type the following:

```
TIME
```

You'll see a message similar to this:

```
The current time is: 10:21:39.03
Enter the new time: _
```

To keep the current time, press **ENTER**. To change the time, type the new time at the cursor, separating hours, minutes, seconds, and hundredths of a second with colons (:) or periods (.). The separator between seconds and hundredths of a second must be a period. The hour is based upon a 24-hour clock. Seconds and hundredths of a second are optional.

You can also specify the time directly after the TIME command, as follows:

TIME 13:30

## SPECIFYING DEVICE INFORMATION (MODE Utility)

### Introduction

The MODE utility prepares MS OS/2 *MULTIUSER* for communication with devices such as parallel and serial printers, modems, and screens.

To use the MODE utility, you must specify the name of the device you want to use followed by a list of arguments. MS OS/2 *MULTIUSER* assigns names to devices as follows:

- LPT1, LPT2, and LPT3 are parallel printers attached to the parallel ports of your computer. If you do not specify a port, LPT1 is the default device name. (PRN can be used in place of LPT1.)
- COM1, COM2, and COM3 are serial devices that are attached to the serial communications ports of your computer. These could include devices such as modems or serial printers.

Before you can set operating arguments for these devices, the device must be installed correctly and the appropriate device driver must be installed. For instructions on how to install device drivers, see Chapter 10, "Using Citrix *MULTIUSER* Device Drivers."

## Setting the Parallel Printer Mode

You can use the MODE utility to set characteristics for parallel printers connected to parallel ports LPT1, LPT2, and LPT3. (PRN and LPT1 can be used interchangeably.) To set the operating arguments for a parallel port, you type MODE followed by the following parallel printer options:

<u>Option</u>	<u>Purpose</u>
LPT <i>n</i>	Specifies the printer number: 1, 2, or 3.
<i>chars</i>	Specifies the number of characters per line: 80 or 132. The default value is 80.
<i>lines</i>	Specifies vertical spacing (lines per inch): 6 or 8. The default value is 6.
P	Specifies that the MODE utility continues to send output to the printer even though a time-out error has occurred. A time-out error occurs when a printer cannot print your output in a previously specified amount of time. You can break out of this process at any time by pressing <b>CTRL+C</b> .

You must set the LPT*n* option; all other arguments are optional.

When changing the parallel printer mode, you are changing the default mode for the system. Users will print in this mode unless they override the default for their loginname using the CHANGE PRNMODE utility (if the spooler is spooling the associated device).



For example, suppose that you want to print to a parallel printer that is connected to your computer's second parallel printer port (LPT2). If you want to print with 132 characters per line and 8 lines per inch, type the following:

```
MODE LPT2: 132,8
```

## Setting the Asynchronous Communication Mode for Custom Async Devices

Select communication ports that do not have a terminal configured for them. You must set the baud option; all other arguments are optional.

The MODE utility is used to configure a specified serial port for communication with external devices such as printers, terminals, and modems. To display the status of a serial device, type MODE followed by the name of the asynchronous port. For example, to see the status of the device that is connected to COM2, type the following:

```
MODE COM2
```

To set the operating arguments for a serial port, type MODE followed by a combination of the following options:

<u>Option</u>	<u>Purpose</u>
COM $n$	Specifies the number of the asynchronous communications (COM) port. The range of valid values depends on your particular computer.
<i>baud</i>	Specifies the transmission rate. Valid rates are 110, 150, 300, 600, 1200, 1800, 2400, 3600, 4800, 7200, 9600, and 19,200. You can use just the first two digits to specify the transmission rate.

<u>Option</u>	<u>Purpose</u>
<i>parity</i>	Specifies the parity. Valid values are N (none), O (odd), E (even), M (mark; the parity equals 1), or S (space; the parity equals zero). The default value is E.
<i>databits</i>	Specifies the number of data bits. Valid values are 5, 6, 7, and 8. The default value is 7.
<i>stopbits</i>	Specifies the number of stop bits. Valid values are 1, 1.5, and 2. If the baud option is set to 110, the default value is 2; otherwise, the default value is 1. If you specify 1.5 stop bits, the databits option must be set to 5.

You must set the baud option; all other arguments are optional.

The options in the following list may be used in any order following the *stopbits* argument.

<u>Option</u>	<u>Purpose</u>
TO= <i>state</i>	Specifies whether infinite time-out processing is enabled (ON) or whether normal time-out processing is used (OFF). The default setting is TO=OFF.
XON= <i>state</i>	Specifies whether automatic transmit-flow control is enabled (ON) or disabled (OFF). The default setting is XON=OFF.
IDSR= <i>state</i>	Specifies whether the input handshake using DSR (data set ready) is enabled (ON) or disabled (OFF). The default setting is IDSR=ON.

<u>Option</u>	<u>Purpose</u>
ODSR= <i>state</i>	Specifies whether the output handshake using DSR (data set ready) is enabled (ON) or disabled (OFF). The default setting is ODSR=ON.
OCTS= <i>state</i>	Specifies whether the output handshake using CTS (clear to send) is enabled (ON) or disabled (OFF). The default setting is OCTS=ON.
DTR= <i>state</i>	Specifies whether DTR (data terminal ready) is enabled (ON) or disabled (OFF), or whether DTR handshaking is enabled (HS). The default setting is DTR=ON.
RTS= <i>state</i>	Specifies whether RTS (request to send) is enabled (ON) or disabled (OFF), whether RTS handshaking is enabled (HS), or RTS toggling is enabled (TOG). The default setting is RTS=ON.

BUFFER=ON | OFF | AUTO

Specifies whether extended hardware buffering is enabled, disabled, or controlled automatically by the device driver. This option applies only to communications hardware or drivers that are capable of extended hardware buffering. For more information about this option, see the *Citrix MULTIUSER Command Reference*.

Suppose you want to use the MODE utility to set up the COM2 serial communications port for 9600 baud, with even parity, 7 data bits, and 1 stop bit. To do this, type the following:

```
MODE COM2: 9600
```

In this example, the parity and number of data and stop bits are not specified because the default values are used. If, on the other hand, you want to set up the COM2 port for 1200 baud, odd parity, 7 data bits, and 2 stop bits, type the following:

```
MODE COM2: 1200,O,,2
```

The arguments you specify when you use the MODE utility to set up your serial port must match the requirements of the device with which you want to communicate.

## Setting the Display Mode

You can use the MODE utility to change the way information is displayed on your screen.

You can use the following options with the MODE utility to set the display mode:

<u>Option</u>	<u>Purpose</u>
<i>display</i>	Specifies one of the following values: 40, 80, BW40, BW80, CO40, CO80, or MONO. For each of these values, 40 and 80 indicate the number of characters per line. BW and CO refer to a color graphics adapter with color disabled (BW) or enabled (CO). MONO specifies a monochrome adapter with a constant display width of 80 characters per line.
<i>rows</i>	Specifies the number of rows of text displayed by the screen: 25, 43, 50, or 60. The adapter type determines which of these values is valid. The initial value is 25; the default value is the last value that you set.



When you type MODE with the display argument, the utility affects the current session only. Suppose your computer uses an 80-character-per-line color graphics adapter (CO80). To change the number of rows of text displayed by the screen to 43, type the following:

```
MODE CO80,43
```

When the MODE command is used with a terminal, the allowed modes depend on the characteristics of the terminal. All terminals support 80x25 modes (MODE CO80,25 or MODE MONO,25); some terminals support other modes. Refer to Appendix D for terminal-specific details.

# CHAPTER 13

## SYSTEM MAINTENANCE

## MAINTENANCE MODE

### Introduction

Maintenance mode is a single user mode of operation. In other words, only one user and one terminal are active. Activities in the system can only be initiated by the user at the active terminal.

In a multiuser system, certain administration activities should take place while the system is quiesced. These activities include:

- Installing the system. When you install MS OS/2 *MULTIUSER*, the system installation programs execute in maintenance mode. This is useful when installing an upgrade or new release over an existing MS OS/2 *MULTIUSER* system by ensuring that no other users are performing activities that could disturb the installation process.
- Running system hardware diagnostics. When you run system hardware diagnostics, you may need to prevent other users from accessing the system devices.
- Performing disk and file management, such as backup and recovery. When you perform disk backup, maintenance mode ensures that no files are being modified during the backup.
- Managing applications, including installation and deletion of the applications. When you install or remove an application in maintenance mode, other users will not be able to run the application or access the application's data.
- Maintaining the system.

## How To Enter And Exit Maintenance Mode

There are three ways to enter maintenance mode.

1. Use the SHUTDOWN utility.

This is the recommended method to enter maintenance mode. The terminal from which SHUTDOWN is invoked becomes the maintenance terminal. Other terminals are quiesced.

To exit maintenance mode, simply restart the system.

2. Use the *CONFIG.SYS* command MAINTENANCE=ON.

This requires that you edit *CONFIG.SYS* for the MAINTENANCE command and restart the system. The system will then come up in maintenance mode.

To exit maintenance mode, edit *CONFIG.SYS* to reset the MAINTENANCE command to OFF and restart the system.

3. Use the MS OS/2 *MULTIUSER* installation diskettes.

This requires that you restart the system with the MS OS/2 *MULTIUSER* install diskette, insert diskette 1 when prompted and press **ENTER**. When the first panel appears, press **ESC** to get the CMD command prompt.

To exit maintenance mode, restart the system after removing the installation diskettes from the diskette drive.



## DISK MAINTENANCE

### Introduction

This section describes the MS OS/2 *MULTIUSER* disk utilities that you use to perform tasks such as displaying information about a disk and fixing disk partition problems.

MS OS/2 *MULTIUSER* provides two utilities that help you maintain your disks. The following list describes these utilities:

<u>Utility</u>	<u>Purpose</u>
CHKDSK	Displays status information about a disk.
RECOVER	Recovers a file or disk that contains bad sectors.

Unlike command and utilities that operate on files and directories, disk utilities are designed to help you maintain floppy and fixed disks.

**NOTE:** Most MS OS/2 *MULTIUSER* utilities work the same in different file systems. In the few cases in which this is not true, MS OS/2 *MULTIUSER* determines the file system you are using and runs the utility for that file system. The utilities that may work differently are:

FORMAT  
CHKDSK  
RECOVER

CHKDSK and RECOVER are discussed in this section. Differences are noted in the text. The FORMAT utility is described in Chapter 12.

## Displaying Disk Information

The CHKDSK utility displays status information about a disk and checks the disk for errors. You should run this utility periodically to keep track of how much space is left on a disk and to display any errors found on the disk. If CHKDSK finds an error, a message describing the error appears on the screen.

To run CHKDSK, type CHKDSK followed by a drive letter or *filename*. CHKDSK always checks a drive; you can also check a file by specifying a *filename*. (You can specify multiple *filenames* by using wildcard characters.) If you specify a filename, CHKDSK checks the drive and also tells you whether the specified file is contiguous. (You can specify a directory path with the *filename*.) If you do not specify a drive, CHKDSK checks the current drive.

For example, to check your fixed disk and save the output in the file *REPORT.DSK* on drive A, type the following:

```
CHKDSK C: >A:REPORT.DSK
```

A typical HPFS CHKDSK status report would look like the following:

```
The type of file system for the disk is HPFS.  
The HPFS file system program has been started.  
CHKDSK is searching for lost data.  
CHKDSK has searched 100% of the disk.  
  30718 kilobytes total disk space.  
  31 kilobytes are in 27 directories.  
  8996 kilobytes are in 248 user files.  
  614 kilobytes are reserved for system use.  
 21077 kilobytes are available for use.
```

A typical status report for a FAT file system would look like the following:

```
The type of file system for disk is FAT.  
The volume label is DAVIDH.  
The volume serial number is E392-1C15.  
  72286208 bytes total disk space.  
  311296 bytes in 7 hidden files.  
  520192 bytes in 125 directories.  
  67895296 bytes in 2443 user files.  
  12288 bytes in bad sectors.  
  3547136 bytes available on disk.  
  
  4096 bytes in each allocation unit.  
  17648 total allocation units.  
  866 available allocation units on disk.
```

The message about hidden files gives you the number of files that are needed by the system but are not shown when you use the DIR command to display a directory list. To discover the names of hidden files, use CHKDSK with the /V option (described later in this section in "Displaying Each File on Your Disk").

The "bytes in bad sectors" message shows you how many of the sectors on your disk are defective. Since the system automatically compensates for bad sectors, there is no reason to worry about them unless the number of bad sectors becomes very large.

The "bytes available on disk" message shows you the number of unused bytes on the disk. You should check this value from time to time, especially if disk space is limited.

MS OS/2 *MULTIUSER* files are stored in a series of units called sectors, which are grouped into larger units called clusters. The CHKDSK utility may report that it has found lost clusters. A lost cluster still exists but the information in it is no longer associated with a file. This could occur for many reasons; for

example, if you turned off your computer while the system was writing to a file. When a cluster is lost, MS OS/2 *MULTIUSER* cannot read, write, or modify the data in it. If part of a file is in a lost cluster, you will not be able to work with the file at all. In addition, lost clusters take up valuable disk space.

If you use the /F option, CHKDSK fixes any lost clusters by writing them to a file. For instructions on how to use the /F option, see the following section.

If you specify files to be checked when you use the CHKDSK utility, you will receive a message that tells you whether the files are stored in contiguous sectors. If a file is not stored in contiguous sectors, the number of its blocks that are stored in noncontiguous sectors is displayed. Although storing files in contiguous sectors is not required, it enables your computer to read from or write to files more efficiently. If many of your files are stored in noncontiguous sectors, you can make all files contiguous by backing up the files on your disk, reformatting the disk, and restoring the files to the disk.

## Fixing Errors on Your Disk

To fix errors that are reported by the CHKDSK utility, use CHKDSK with the /F option. Because this option must lock the entire partition during its processing, it is recommended that you use this only in maintenance mode (see the previous "Maintenance Mode" section). If you are running CHKDSK on a floppy disk, this isn't necessary.

In addition, if you need to fix drive C and you started your system from drive C, you must restart the system from drive A.



You can use the copy of the CHKDSK utility that was put on your start-up disk when you installed MS OS/2 *MULTIUSER* to fix errors on any disk except your start-up disk. To fix errors on drive C, do the following:

1. SHUTDOWN the system.
2. Insert the MS OS/2 *MULTIUSER* Install disk into drive A and turn on your computer.
3. Insert disk 1 when prompted and press **ENTER**.
4. When the first panel appears, press the **ESC** key. This takes you to the command prompt.
5. Type `CHKDSK C: /F` and press the **ENTER** key.

The CHKDSK utility attempts to recover lost clusters and places their contents into files that have the *FILEnnnn.CHK*, where *nnnn* is a sequential number assigned by MS OS/2 *MULTIUSER*. The recovered files may or may not be usable, depending upon their condition.

If the files with lost clusters are ASCII files (data files), you may be able to recover the lost data. If you find that there is text missing from an ASCII file, run CHKDSK with the `/F` option, then do the following:

1. Use the `TYPE` command to look at the contents of each of the *.CHK* files.
2. If the missing text appears in one of the *.CHK* files, use a text editor to copy this text back into the original file.

In MS OS/2 *MULTIUSER*, some applications attach additional information, such as author, application type, and file history to your files and directories. This type of information is called extended attributes and is used by other applications, the file system, or the operating system itself. If CHKDSK finds errors in extended attributes, the recovered extended attributes are placed in files named *EA#####.CHK*, where *#####* is a sequential number assigned by MS OS/2 *MULTIUSER*.

You can look at files that have extended attributes by using the TYPE command, as described previously. If, after using the TYPE command, you can identify which of your files the extended attributes belong to, you can join the extended attributes to the files by using the EAUTIL utility. For information about using EAUTIL, see the *Citrix MULTIUSER Command Reference*.

Note that the HPFS CHKDSK places these *.CHK* files in a new directory named *FOUND*, located in the root directory. It also creates an additional *DIR#####.CHK* file for recovered directories and their contents.

## Displaying Each File on Your Disk

Use the /V option if you want the name of each file on the disk to be displayed as it is being checked. The /V option causes the CHKDSK utility to display all files, including hidden files. (However, the hidden files are not flagged in any way.) Discovering which files are hidden files requires some checking. If you need to know the names of the hidden files, follow these steps:

1. Use the CHKDSK utility without the /V option. This tells you the number of hidden files; however, it does not tell you the names of the files.

2. Use the CHKDSK utility with the /V option to produce a list of all of the files on a particular drive.
3. Use the DIR command to produce a file list that does not show any hidden files.
4. Compare these two lists. Any file that is in the CHKDSK /V list but is not in the DIR list is a hidden file. You can use the number produced in step 1 to make sure you have found every hidden file.

Since CHKDSK /V displays filenames left-aligned, you can easily use the SORT utility to sort filenames and directories alphabetically. The SORT utility is described in detail in Chapter 4, "Using Citrix *MULTIUSER* Utilities" in the *Citrix MULTIUSER User's Guide*.

## Recovering a File From a Damaged HPFS Disk

The HPFS RECOVER utility works slightly differently from the FAT RECOVER. This section describes the behavior of HPFS RECOVER.

You can use the HPFS RECOVER utility to recover a file that has been stored in a bad sector on your fixed disk. This utility can recover individual files only; it cannot recover an entire disk. To recover a file, type RECOVER followed by the name of the file to be recovered.

The RECOVER utility gives a recovered file its original filename, if possible. If the original filename cannot be restored, recover creates the file *FILEnnnnn.REC* in the directory where the original file existed. (The characters *nnnnn* specify a sequential number assigned by MS OS/2 *MULTIUSER*.)

If the sector(s) containing a damaged file cannot be recovered, the RECOVER utility retains the original file size by filling the unrecovered portion of the file with zeros.

If no errors are found in a file's extended attributes, the RECOVER utility preserves the extended attributes. If errors are found in the extended attributes, the RECOVER utility recovers as many of the attributes as possible and stores them in a file name *EAnnnnn.REC*. (The characters *nnnn* specify a sequential number assigned by MS OS/2 *MULTIUSER*.)

## Recovering a File From a Damaged FAT Disk

If a sector on your FAT disk is damaged, you can use the RECOVER utility to try to recover either just the file that has been written to the damaged sector or the entire disk.

Before using this utility, be sure that you have backup copies of all the files on your disk. Then try to restore your files by using the RESTORE utility. If this fails, use the RECOVER utility on one file at a time. Use RECOVER on a disk only if the entire disk is unreadable.

To recover one file, type RECOVER followed by the name of the file you want to recover. MS OS/2 *MULTIUSER* reads the file sector by sector, skipping the bad sectors, and labels any bad sectors so that no data is written to them in the future.

To recover an entire disk, type RECOVER followed by the drive letter.

The RECOVER utility does not work on a network from a remote workstation.



When you use the RECOVER utility on a disk, no other program can access that disk while the utility is running. This prevents other programs from modifying the disk while RECOVER is modifying it. Likewise, if a program is already using a disk, you cannot run RECOVER on the disk from which you started MS OS/2 *MULTIUSER*, nor can you run RECOVER on the disk from which you started RECOVER.

If you need to run RECOVER on the disk that contains the RECOVER utility, you need to copy the *RECOVER.COM* file to another disk (either a floppy or fixed disk) and run the utility from there.

When recovering a single file, the RECOVER utility attempts to recover the extended attributes of the file. If it cannot recover the complete extended attribute information, what it can remove is stored in a file named *EAnnnnn.REC*. (The letters *nnnn* designate a sequential number assigned by the operating system.) You can read this file by using the TYPE command. The recovered file will no longer have extended attributes.

When recovering an entire drive, all of the files lose their extended attributes. The RECOVER utility stores the extended attributes in a file named *FILEnnnnn.REC*. You can read this file by using the TYPE command, but there is no way to tell which files the extended attributes originally belonged to.

## BACKING UP AND RESTORING THE SYSTEM CONFIGURATION

### Introduction

Your MS OS/2 *MULTIUSER* system configuration should be backed up occasionally in order to preserve the system profiles describing your installation. Since the profiles are

static and change only when the Administrator makes a configuration change, it is appropriate to do a backup shortly after making significant configuration changes, such as adding a number of new terminals.

## Critical System Directories

Backing up system data involves saving the contents of critical directories. The following describes the basic MS OS/2 *MULTIUSER* directory structure.

\OS2	<p>This directory and its subdirectories contain almost all of the programs (EXEs, DLLs, and device drivers) that make up the MS OS/2 <i>MULTIUSER</i> product. Some programs are found in the root directory (\). These are used during the boot sequence. The contents of this directory is established during system install.</p> <p>This directory can be recreated by re-installing the product.</p>
\SYSCTX	<p>This directory (and its subdirectories) contains all of the system profiles and all of the security attributes which describe security on resources other than files and directories. This data is maintained and updated using MS OS/2 <i>MULTIUSER</i> utilities such as CONFIG.</p> <p>This data must be backed up after significant configuration changes.</p>
\USR	<p>This directory is the parent directory for all user home directories. For each <i>username</i> configured in the system, a subdirectory under this directory exists to house user oriented system data and to provide a user home directory.</p>

The file data in this directory structure is basically user data. The procedures for dealing with this will depend on the installation.

**\CONFIG.SYS** This is a file in the root directory that describes the basic system configuration. This file is created during system installation. Many applications require updating this file during their installation.

This file should be copied to a backup diskette after any changes are made.

## **How to Backup the System Profiles**

Use the BACKUP utility to backup the system profiles. This will handle the long filenames, and will backup all of the security attributes. You should make this backup copy while no one is actively making changes to the profiles.

It is recommended that you do this BACKUP while in maintenance mode.

To make a backup using drive A, execute this command:

```
BACKUP C:\SYSCTX A: /S
```

Refer to the description of the BACKUP utility for details on how this utility operates.

Again, it is important that the BACKUP utility be used for this purpose because it preserves the security attributes and it handles long filenames.

## Restoring the System

The specific steps in restoring a system depend on its current state. These steps assume that all data on the fixed disk has been lost.

1. Reinstall MS OS/2 *MULTIUSER*.  
This will recreate the \OS2 subdirectory and establish a base system as if this was the first install.
2. Reboot the newly installed system and login using the administrator ID established during the install.
3. SHUTDOWN the system and go into maintenance mode.
4. Restore the system profiles from your backup diskettes using the following command:

```
RESTORE A: C:\SYSCTX /S
```

5. Re-establish the \USR directory. This is done manually using the make directory command (MD) for each *username* on your system. You also need to set up the security controls using CONFIG ACCESS.
6. Using your backup copy of *CONFIG.SYS*, edit the new *CONFIG.SYS* to add any changes you may have made. This step may not be necessary depending on whether or not you need to reinstall your applications.
7. Reinstall applications as necessary.
8. Reboot.



## BACKING UP SYSTEM FILES THAT ARE OPEN

The BACKUP utility does not make copies of files if the files are already open or if they are in use by another program. For example, if you run BACKUP, files such as the *SWAPPER.DAT* are not backed up because they are being used by MS OS/2 *MULTIUSER*. To back up these files using the BACKUP utility, follow these steps:

1. SHUTDOWN the system.
2. Insert the MS OS/2 *MULTIUSER* Install disk into drive A of the system console and turn on the console.
3. Insert disk 1 when prompted and press **ENTER**.
4. When the first panel appears, press **ESC**. This takes you to the command prompt.
5. Start the BACKUP utility. You can make backup copies of individual files or of the entire start-up disk.

To restore these files to the start-up disk, run the RESTORE utility from drive A in the same way as described in the preceding steps.

It is not normally necessary to BACKUP the files described in this section; they are typically recreated at boot time or when you restore the system.

**NOTE:** The BACKUP utility does not make backup copies of the file *CMD.EXE* or of hidden system files.

# CHAPTER 14

## SYSTEM EVENT LOGGING

## INTRODUCTION

MS OS/2 *MULTIUSER* contains an event manager that tracks and logs certain system events. This log is used primarily for maintaining a Security audit trail. However, other system events can be logged to help in problem determination.

The event manager uses the general MS OS/2 *MULTIUSER* system log facility.

## GENERAL SYSTEM LOGGING FACILITY

MS OS/2 *MULTIUSER* provides a generalized system logging facility. Event management is one type of use for this; others are device error logging and communications alert generation. However, the bulk of the log messages are MS OS/2 *MULTIUSER* system events, such as security audits.

The generalized logging facility is controlled by the LOG *CONFIG.SYS* parameter. (The SYSLOG utility is also part of this general facility, but is not discussed here.)

## ENABLING AND DISABLING THE SYSTEM LOG FACILITY

During a standard system install, the logging facility is always enabled. This is done by placing a LOG=ON command in *CONFIG.SYS*. The LOG=ON command starts the logging process and sets aside a logging buffer in memory and on disk. System install uses the default values for this command; however, an Administrator may wish to change some of the parameters such as buffer size. This is described in the *Citrix MULTIUSER Command Reference*.

System logging can be disabled by placing LOG=OFF in *CONFIG.SYS*, and rebooting. However, if this is done, no security auditing will take place and no event logging will occur. With LOG=OFF, no other facilities described in this chapter will work.

## MANAGING THE LOG FILES

All data being logged is written to a log file. This file remains open during all system operation and will grow until the disk space limit specified on the LOG= command is reached. When this occurs, a popup is generated and logging is suspended.

MS OS/2 *MULTIUSER* provides a mechanism to "swap" this file with a new file so that this file can be moved to a backup diskette and deleted, or so it can be used to generate reports. The EVENTS utility is used to swap the log file.

It is recommended that the log file be swapped on a regular basis rather than encountering the condition where the log file fills up. Once it fills up, logging is suspended and data is lost. If this occurs, the system must be restarted to re-establish logging.

### Naming the Log File

All data being logged is written to a log file called \OS2\SYSTEM\LOG0001.DAT

A new log file is created only when the Administrator requests a swap using the EVENTS utility.



When a new LOG0001.DAT is created, the current file is renamed as follows:

LOG*mmdd*.###

where *mmdd* is the month and day the new file is created and ### is the next available sequence number for file LOG*mmdd*.

## SELECTING EVENTS TO LOG

Events are identified by event category and event ID. The event category is a major grouping of types of events. Each category is described by a single letter. Categories are:

- A - Security access audit
- E - Errors
- T - Terminal subsystem events
- M - Multiuser events
- R - Resource events
- P - Profile update events
- S - Security Secondary audit

The event ID uniquely numbers an event within a category.

Selection of events is done in two ways:

### ■ Using CONFIG SYSTEM

This utility allows the Administrator to establish a static event set that should be logged at all times. This event selection is placed in the system profile and is re-established on each reboot. This is the best way to ensure that the desired events are being logged.

### ■ Using CHANGE EVENTS

This utility will dynamically change the event selection. This will not change the event selection in the system profile but overrides the selections during system operation. Any changes made with this utility are lost on the next system boot.

The security audit events are those generated by the security system. There are a number of ways to generate these events, many of which are under the control of the System Administrator. The various security events become different event IDs in the context of the event manager. A detailed discussion of controlling audits is provided in Chapter 7, "Configuring Security."

## FORMATTING THE LOG DATA

The data in the log file is kept in an internal format. To generate a readable report, you must run the EVENTS utility. This utility allows you to extract all, or selected events, and redirect the formatted data to a file for later printing or editing.

A typical scenario would be the following:

1. Close out the current log with EVENTS /SWAP.
2. Display the \OS2\SYSTEM directory to determine the name of the most recent log.
3. Format the log data with the following:

```
EVENTS LOG0802.001 >LOG0802.OUT.
```

4. View the LOG0802.OUT file using System Editor.

## A CAUTION ABOUT SECURITY AUDITING

Security auditing uses the facilities of system event logging. There are a number of ways to control security auditing, as described in Chapter 7, "Configuring Security." However, security auditing is merely a single event category to the event manager and can be enabled or disabled at this level of control.

# CHAPTER 15

## SPOOLER OPERATION



## INTRODUCTION

The MS OS/2 *MULTIUSER* SPOOL utility allows you to view the status of the system printing activity and control the operation of print queues and print jobs within queues.

## SPOOL UTILITY PANEL

Type SPOOL and press **ENTER** at the command prompt to start the full screen SPOOL utility. The SPOOL utility panel shows all the current print queues in the system and all the print jobs in each queue. An example of the SPOOL utility panel follows:

Spool Utility			
Queue	Job	Setup	Exit
--> MARKB.ENGINEERING			
Name	Job ID	Size	Status
-----			
LAZER JET	3 Job(s)		Queue Active
└ ALC.ENGINEERING	9	11280	Printing
└ ROGERR.CEO	13	523	Queued
└ ANDYS.ENGINEERING	14	0	Spooling
DOT MATRIX	1 Job(s)		Queue Held
└ GREGG.ENGINEERING	16	5423	Queued
Enter=Details    F1=Help    F3=Exit    F5=Refresh    F10=Actions			

## Refresh the SPOOL Utility Panel

The **F5** key can be used to refresh the screen at any time to get the current status of the printing activity.

Refreshing can also be done automatically every  $x$  seconds, where  $x$  is configurable. Refer to Chapter 8, "Configuring Printing and Spooling" for more information on the SPOOL utility.

## Detail Zoom the SPOOL Utility Panel

Press the **ENTER** key to display details on the highlighted item on the main menu panel.

## PRINT QUEUE CHANGES

**NOTE:** If the highlight bar is not on a queue, the Queue Menu pulldown items are not selectable.

## Hold a Print Queue

To hold a queue means to prevent it from sending jobs in the queue to a printer. If a job is currently printing, it is not affected by the hold. Use the following steps, to hold a print queue.

1. Highlight the print queue you wish to hold.
2. Press **F10** to go to the action bar.
3. Select Queue on the action bar.
4. Select Hold Queue.

**NOTE:** This is not selectable if a queue is held already.

## **Release a Print Queue**

Releasing a print queue will continue printing all jobs in that queue. Use the following steps, to release a print queue.

1. Highlight the print queue you wish to release.
2. Press **F10** to go to the action bar.
3. Select Queue on the action bar.
4. Select Release Queue.

**NOTE:** If the print queue is not held, this is not selectable.

## **Cancel All Jobs in a Print Queue**

Use the following steps to purge all jobs in a print queue.

1. Highlight the print queue you wish to purge.
2. Press **F10** to go to the action bar.
3. Select Queue on the action bar.
4. Select Cancel All Jobs in Queue.

## **PRINT JOB CHANGES**

**NOTE:** If the highlight bar is not on a print job, the Job Menu pulldown items are not selectable.

## **Cancel a Job**

Use the following steps to remove a job from the print queue.

1. Highlight the job in the print queue you wish to cancel.
2. Press **F10** to go to the action bar.
3. Select Job on the action bar.
4. Select Cancel Job.

## Print Job Next

Use the following steps to make the specified job the next job to be printed. If the job is held, it is released and put at the head of the queue.

1. Highlight the job in the print queue you wish to print next.
2. Press **F10** to go to the action bar.
3. Select Job on the action bar.
4. Select Print Job Next.

## Start the Job Again

Use the following steps to stop the print job that is printing and start printing again from the beginning:

1. Highlight the job in the print queue you wish to stop and print again.
2. Press **F10** to go to the action bar.
3. Select Job on the action bar.
4. Select Start Job Again.

**NOTE:** If the job is not printing, this item is not selectable.



## **Hold the Job**

Use the following steps to pause the print job so that it will not be printed until released. If the job is printing, then it will not be held.

1. Highlight the job in the print queue you wish to hold.
2. Press **F10** to go to the action bar.
3. Select Job on the action bar.
4. Select Hold Job.

**NOTE:** If the print job is already held, this item is not selectable.

## **Release a Job**

Use the following steps to release the hold on a job so it can be printed.

1. Highlight the job in the print queue you wish to release.
2. Press **F10** to go to the action bar.
3. Select Job on the action bar.
4. Select Release Job.

**NOTE:** If a job is not held, this item is not selectable.

# CHAPTER 16

## PROBLEM DETERMINATION

## TERMINAL DOESN'T WORK

The following is a list of things to do and to check if the terminal fails to bring up the login prompt or the first user program.

- Verify that the terminal power cord is plugged into the wall outlet on one end and the terminal on the other.
- Verify that the brightness and contrast controls on the terminal are adjusted so that you can see the screen.
- Press the **CTRL** or **ALT** key (actually any key will do) to ensure that the screen saver has not turned the screen off. Pressing a key will make the screen become visible if the screen saver turned it off.
- Verify that the serial cable is connected into the back of the terminal on the terminal side and connected into the host computer on the other side.
- Press the **ENTER** key to see if the login screen appears. Some terminals have varying power on delay requirements that are not always met. Pressing **ENTER** causes the login prompt to be redisplayed.
- Turn the terminal off and then back on to reset it and listen for the beep (most terminals beep when they've finished their power on diagnostics).
- Verify that the terminal keyboard cable is properly seated.
- Use the **QUERY** utility to determine the status of the malfunctioning terminal. If the terminal is present, use the **KILL** utility to kill all the logins that are running at the terminal.

- RESET the terminal by executing the RESET TERMINAL utility.
- Verify all configuration parameters that are set up using CONFIG TERMINAL match the configuration parameters that are set up at the terminal. Pay particular attention to the parity, baud rate, stop bits and data bits parameters.

**NOTE:** Refer to Appendix D, "Setting Up Specific Terminal Types," for more information on the terminals that are supported.

- Verify that the connection type parameter is set up properly for the cabling used between the terminal and computer. For three wire cable, "connect on first character" and "disconnect on break" should be used. For four wire null modem cable, "connect on DSR" should be used. For other non-standard cables, refer to Appendix E.
- If the terminal appears to generate spurious keystrokes, verify that the XON/XPC protocol is set to XPC protocol and that the terminal keyboard is configured to send PC scan codes.
- If all terminals on a given terminal subsystem are not functional, verify that the MS OS/2 *MULTIUSER* Async device driver is installed properly. Verify that the parameters that are provided on the "DEVICE = " statement in the *CONFIG.SYS* file are correct. It is very important that the Micro Channel async device driver only run on a Micro Channel system and an Industry Standard Architecture device driver only run on an Industry Standard Architecture system. For unintelligent serial terminal subsystems, the Industry Standard Architecture device driver is named COM01.SYS, and the



Micro Channel device driver is named COM02.SYS. The intelligent terminal subsystem manufacturers typically have both Micro Channel and Industry Standard Architecture offerings and thus have both types of device drivers.

- Again, if all terminals on a given terminal subsystem are not functioning properly, verify that the switches on the adapter board are set properly for Industry Standard Architecture adapters. For Micro Channel adapters, verify that the proper setup parameters are set using the setup program that is provided with your Micro Channel system. Pay particularly close attention to the interrupt level parameter and the memory map location parameters. In Industry Standard Architecture systems, it is required and in Micro Channel systems, it is recommended that the serial terminal subsystem have exclusive ownership of an interrupt level. When choosing a location in memory where the adapter's memory map is to be addressed, it is also particularly important to understand the memory caching that is provided by the host computer. The memory locations where the adapter memory is placed must be made noncachable.

## CANNOT ACCESS DISK BECAUSE OF SECURITY

There are many instances when the user is not allowed to perform normal functions due to security restrictions. The usual error message received is "Access Denied ...". Listed below are the most common occurrences of a security violation that produce the error message of "Access Denied ..."

- When a user tries to MKDIR (make directory) and does not have Create (C) permission on the parent directory.
- When a user tries to RMDIR (delete directory) and does not have Delete (D) permission on the target directory.
- When a user tries to "execute" a file and does not have Execute (X) permission on the file.
- When a user tries to "read" a file and does not have Read (R) permission on the file.
- When a user tries to CHDIR (change directory) user does not have at least Read (R) permission on the directory.
- When a user tries to modify a file via an editor, COPY, MOVE, XCOPY, and so forth, without Write (W) permission.
- When a user tries to modify file or directory security attributes and does not have modify Attributes (A) permission.
- When a user tries to access a disk and the user does not have Use (U) permission.

**NOTE:** The user may have to RESERVE the device prior to using it.

- When a user tries to execute a function that is not allowed, because the user does not have the proper authority to do so. The user may try to execute a function that requires one of the following:
  - Query Self
  - Query Others
  - Modify Self
  - Modify Others

As an example, a typical user may be given the authority to Query and modify his or her *password*. Thus, if a user tries to query or modify another's *password* through the PASSWORD utility, the user gets the "Access Denied" message.

- When a user tries to RESERVE a device that the user does not have authority to reserve or someone else has already reserved it, the "Access Denied" message is displayed. For example, if a user wants to COPY something to a diskette and does not want anybody else to copy to that diskette, the user attempts to RESERVE the drive letter. If someone else has the drive reserved, the user gets the Access denied message.

To correct the access problem follow these steps:

1. Determine if the intended access is valid and is not a breach of security.
2. Use the CONFIG ACCESS utility to give an *accessname* the proper permission to accomplish the task.
3. Retry the task.

## FORGOTTEN PASSWORD

A user with Administrator class privileges can change any other user's password without having to know the user's current password. To change a user's password, type the following (on one line):

```
PASSWORD username[.groupname]  
/PASSWORD:/newPW/newPW
```

The password for user *username[groupname]* will be changed to *newPW*. You can also change the password using the full screen configuration utilities and leaving the current password field empty. Refer the section in Chapter 5, "Changing a Password" for more information.

## LAST ADMINISTRATOR DELETED

Complete the following steps if the last user with Administrator class privileges has been deleted or the account has expired. You will need to restart the system which will terminate all existing logins.

To create a new user with Administrator class privileges:

1. Insert the MS OS/2 *MULTIUSER* Install disk in drive A:, restart your computer, insert Disk 1 when prompted and press **ENTER**, then press **ESC** at the first panel.
2. Type C: at the command prompt and press **ENTER** to make drive C the current drive.
3. Display the Profile Configuration menu by entering the command CONFIG PROFILES.
4. Move the selection bar over the groupname that you want the user to be a member of.
5. Press **F10** to go to the action bar, select Profile, then select New.
6. Type the name of the new user and press **ENTER**. The new user is created.
7. Move the selection bar over the new username that was just created and press **ENTER** to edit the profile.



8. Change the user's class level to Administrator and press F4 to file the changes.

The new user now has Administrator class privileges and can be used when you restart MS OS/2 *MULTIUSER*.

## CANNOT DELETE A FIXED DISK PARTITION

If you cannot delete the extended partition on a fixed disk, make sure that you have deleted all the logical drives that are in it. If you cannot delete the primary partition on a fixed disk, make sure that there is no extended partition on that disk. If there is, then you must delete it.

If you still cannot delete the partition, then it may be a partition that is not related to MS DOS or MS OS/2. FDISK cannot be used to delete partitions that are not related to MS DOS or MS OS/2. You must use the operating system that was used to create the foreign partition in order to delete the partition.

## USERS FAILING DUE TO LACK OF SYSTEM RESOURCES

### **Users Cannot Login Due to Lack of Resources**

Resolutions to Resource Management related login difficulties follow:

- The user trying to log in has a minimum limit that is set too high. The solution is to reduce the minimum limit.
- One or more users have logged into the system and have consumed all available resources. This could only happen if the user has a relatively high maximum limit

while system resources are low. The solution would be either to increase the amount of resources available or set the user's maximum limit to a lower value.

Some of the ways you can increase the availability of resources follow:

Memory	Free disk space, terminate unnecessary applications, buy more memory, reduce the size of any VDISK's.
Threads	Terminate unnecessary applications, increase the amount of threads in <i>CONFIG.SYS</i> (see <i>THREADS=</i> ).
System Semaphores	Terminate unnecessary applications.
File Handles	Terminate unnecessary applications

## Out of Resource Hard Error Popups

There are two types of out of resource errors that the user may experience:

### User Maximum Limit Reached

This error will occur if a user's application attempts to allocate resource when the user has already reached his or her maximum resource limit. The error will come in the form of a hard error popup that gives the user the option of returning the error to the program or retrying the operation.

An Administrator or Operator can increase the user's maximum limit temporarily via the `CHANGE LIMITS` command. This increase in resources will only be in effect during the current login. If the user then retries the operation, it will succeed. Once the user logs out and logs back in again, the resource is returned to its original limit.

### Request Would Interfere With Other Users' Minimum Guaranteed Limits

This error occurs once a user has exceeded his or her own minimum limit. However, there are not enough resources to allow additional allocations due to other users' guaranteed minimum limits.

There are several alternatives that can help the user out of this problem; some of them are specific to a given resource type:

- Reduce the load on the system. If the user is running applications that are not needed, the applications should be shut down to free up the resources they are using. An Operator or Administrator could kill such processes for the user.
- Other users' minimum limits could be reduced. This reduces the amount of memory reserved for these users. Use the `CHANGE LIMITS` utility to do this if you want the change to be for this login only. If you want the changes to be permanent, use the `CONFIG USER` command.
- For virtual memory resource errors: If disk space can be freed up, the amount of virtual memory could be increased to allow further memory allocations.

## AN APPLICATION DOES NOT RUN PROPERLY

### Security Considerations

#### Popup Indicating Error Code SYS0005

A program (.EXE or .DLL) that is trying to access a secure Application Program Interface (API) generates an error popup when it is loaded. The popup provides the name of the API that the program is attempting to access and shows the load error to be "SYS0005, Access Denied."

If the program is allowed to make the secure function call, it must be "registered" and the access control for the failing API should be updated with the program's aliasname. See Chapter 7, "Configuring Security."

#### Access Denied During Load

A program is trying to load a .DLL or an .EXE but the user or program does not have execute access to the .DLL or .EXE. An extensive discussion of the load process is described in Chapter 7, "Configuring Security."

#### Access Denied During Application Execution

A program is attempting to access a resource but the user or program does not have required permissions. Review the audit log to see what resource access was denied. See Chapter 7, "Configuring Security," and Chapter 11, "Application Installation."

### System Global/User Login Global Considerations

These problems could be related to how resources are managed in MS OS/2 *MULTIUSER*. If the application that doesn't run is a Client/Server type application or already has built in multiuser functions, it may need to be marked system global.



However, very few applications require system global resources. If you have an application that does not work correctly, some or all of its *.EXE* and *.DLL* files may need to be marked System Global.

See Chapter 11, "Configuring Applications," for more information.

## Time Critical Priority Considerations

Applications must be registered to gain access to Time Critical Priority. Problems can occur if a program needs Time Critical Priority:

- **Data Loss**  
High speed communications applications require immediate dispatching to keep from losing data.
- **Poor Performance**  
Some applications may experience poor performance if they cannot get Time Critical Priority.

If data loss or poor performance seems to be occurring, then the program may need to be registered for Time Critical access using the REGISTER and CONFIG ACCESS utilities. The EVENTS utility can be used to find out if the Time Critical Priority has been denied to the program. For more information on registering programs for Time Critical Priority, see Chapter 11, "Configuring Applications." The REGISTER, CONFIG ACCESS, and EVENTS utilities are described in the *Citrix MULTIUSER Command Reference*.

# APPENDIX A

## *README* FILES

### WHAT IS A *README* FILE

*README* files are installed on your system during the MS OS/2 *MULTIUSER* installation procedure. These files contain information that is intended to supplement the publications that come with MS OS/2 *MULTIUSER*. The first *README* file should be read so you can determine whether or not there is any information contained in the *README* files that you are interested in. The procedure contained in the following section assumes that MS OS/2 *MULTIUSER* has been installed successfully and you have successfully logged on to the console or a terminal and can get a command prompt.

### HOW TO READ OR PRINT A *README* FILE

The procedure to be able to view or print the first *README* file is to enter *README* at the command prompt. Without any input parameters, entering *README* displays information on all the other *README* files that have been installed and how to read or print them.



# APPENDIX B

## DIFFERENT WAYS TO SET UP USERS AND SYSTEM STARTUP

The System User/Terminal Startup Options section of Chapter 1 summarizes the different user and terminal startup options. In addition, Part 3 of this book explains the tremendous amount of flexibility in how users, groups, and terminals can be set up and how security is applied to the system resources. This Appendix discusses some possible real life scenarios and how MS OS/2 *MULTIUSER* would be set up to support the scenarios.

### SCENARIO 1

A number of order entry clerks all run the same order entry program. For security reasons, each clerk must login before using a terminal. The only program that the clerks use is the order entry program.

The system is configured as follows. Each order entry clerk is given a unique *username* and *password*. They are all placed in the same default group called CLERK. Clerks are placed in the Guest security class. Every clerk's initial working directory is set to the directory that the order entry program is in and every clerk's initial program is set to the order entry program. Every terminal is set up to have a login prompt. A clerk can login at any terminal and the system will still be able to identify who it is.



The order entry program can automatically determine which user it is interacting with by using the USERNAME environment variable. Special security setup is not required because the only program that each clerk can run is the order entry program. The order entry program was written so that the associated users do not have access to the general system.

Each clerk's user limits are set up to allow only one screen session. This prevents the terminal Session Create hotkey sequence from creating a new session for the clerk because then the clerk could then perform a different function (which is not desired).

## SCENARIO 2

A number of terminals are used to run an order entry program. The program comes up automatically when the terminals are powered on. Login security is not required. The identity of the operator who enters the order is not important, but the identity of the terminal that the order was entered on is important to the order entry program. The only program that can be run from these terminals is the order entry program.

The system is configured as follows. Every terminal is configured to autologin under a different *username* identification but the same default group. Each *username* is configured to have the same initial working directory and the same initial program. The initial program is the order entry program. There is a one-to-one correspondence between the terminalname and the username that is chosen for the autologin.

The order entry program can determine which terminal is interacting with it by using the USERNAME environment variable. Special security setup is not required because the only program that can be run from the terminal is the order entry program. The order entry program is written so the user does not have access to the general system.

Each terminal's Session Create hotkey sequence is configured to be null so another session that could be used to run another program cannot be created.

## SCENARIO 3

In addition to the clerks in Scenario 1, there are also supervisors. These supervisors need to be able to login and execute a spreadsheet, a word processor, and a supervisor program. These are the only programs that the supervisors should be able to run. The supervisor program checks certain environment variables to determine certain parameters automatically. It does not derive these parameters from the identity of the user.

The system is configured as follows. The clerks are configured the same as in Scenario 1. In addition, each supervisor is given a unique *username* and *password*. They are all placed in the same default group called SUPER. Supervisors are placed in the User security class. Each supervisor's initial default directory is set up to be the supervisor's home directory. Each supervisor is given a Program Selector with the three authorized programs already set up. There is a CONFIG.USR file for every supervisor. The file is used to initialize certain environment variables that the supervisor program requires. Every terminal is set up to have a login prompt.

Security is set up so users in the SUPER group can only execute the three programs in their Program Selector. They may not execute the order entry clerk program or any of the programs that the MANAGER group can execute in scenario 4. The SUPER group cannot access any of the order entry clerk data or any of the MANAGER group data either.

## SCENARIO 4

In addition to scenario 3 there are other users who are part of the MANAGER group. These users have complete access to everything in the system and also have access to the system administration functions. Some programs that these users want to execute are started automatically when they login.

The system is configured as follows. In addition to what is required for Scenario 3, each manager is given a unique *username* and is put in the MANAGER group. Managers are placed in the Administrator security class. Each manager has a STARTUP.CMD file setup to automatically start certain programs. Each manager's Program Selector is also set up to contain all the programs that the manager may want to run.

This Appendix has shown how one series of real life scenarios can be accommodated by the flexibility found in the MS OS/2 *MULTIUSER* configuration capabilities.

# APPENDIX C

## INSTALLING ASYNC PORTS FOR Citrix *MULTIUSER*

This Appendix discusses the various options for adding async (serial) ports to an MS OS/2 *MULTIUSER* system. These ports can be used to support multiple terminals or dial-in ports. Several different cards are presented as examples. Note that any multiport asynchronous cards with MS OS/2 driver support can be added to an MS OS/2 *MULTIUSER* system. Some manufacturers may have special drivers available specifically for MS OS/2 *MULTIUSER*. Refer to the manufacturer's documentation for further details.

### GENERAL CONSIDERATIONS

Most personal computers only contain one or two async ports. In order to support more users on an MS OS/2 *MULTIUSER* system, more async ports must be added with multiport async cards.

Two types of async port cards can be used. The first kind is a conventional single or dual port card. This is an unintelligent card using a National Semiconductor 16450 (or equivalent) which would be used to add up to two async ports as COM1 and/or COM2. This is a low-performance configuration. The other kind of async card is an intelligent multiport card. This is a high-performance interface containing its own microprocessor and memory. An intelligent multiport async card "relieves" the master processor of some of the port-related overhead, thus speeding up the system.



If you intend to use the built-in async ports, make sure that the system is configured to properly enable the ports. Most systems can only support two conventional async ports (COM1 and COM2). Refer to the computer's hardware manual for details.

## UNINTELLIGENT ASYNC CARD SETUP

This section gives setup information for several selected unintelligent async adapters. This is not a complete list, but any adapter that uses the National Semiconductor 16450 chip or equivalent that is compatible with MS OS/2 *MULTIUSER* should work. Refer to the manufacturer-supplied literature for setup details.

Whether you use only the planar port or also use an additional unintelligent async adapter, you will need to add the appropriate driver inclusion statement to your *CONFIG.SYS* file for the COM driver to be properly enabled. This statement is "DEVICE=C:\OS2\COM01.SYS" for Industry Standard Architecture (ISA) machines and "DEVICE=C:\OS2\COM02.SYS" for Micro Channel machines.

### **Boca Research IO/AT Board**

A Boca Research IO/AT Board (ISA bus) will add one or two ports to the machine, depending on how many ports are already present. This configuration will allow you to connect two terminals to your system. This is an unintelligent, low-performance async board.

Some sample jumper settings to use the Boca Board are presented here. Place the board on a flat surface with the chips on the top surface and the card edge connector at the bottom, the DB-9 and DB-25 connectors to the right.

- The "P" jumper is the parallel port jumper. Place a jumper between the "LPT1" pin and the center pin to configure the port as LPT1. Place a jumper between the "LPT2" pin and the center pin to configure the port as LPT2. Remove this jumper to disable the parallel port on the card.
- The jumper block to the right of the "P" jumper and the "IRQ3" and "IRQ4" jumpers are used to set up the serial ports. There are two async ports on the Boca Board. One or both of the ports can be used.
- To configure the first port as COM1 and the second port as COM2, set the jumpers as follows:
  - a) Column 1: jumper between the topmost two pins ("A")
  - b) Column 2: jumper between the bottom two pins ("B")
  - c) Column 3: no jumper
  - d) Column 4: no jumper
  - e) IRQ3 jumper: jumper between the "B" pin and the center pin
  - f) IRQ4 jumper: jumper between the "A" pin and the center pin

Use this configuration if no async ports are built into the computer.

- To configure the first port as COM2 and not use the second port, set the jumpers as follows:

- a) Column 1: jumper between the bottom two pins ("A")
- b) Column 2: no jumper
- c) Column 3: no jumper
- d) Column 4: no jumper
- e) IRQ3 jumper: jumper between the "A" pin and the center pin
- f) IRQ4 jumper: no jumper

Use this configuration if one async port is already installed in the computer as COM1.

The 9-pin connector on the Boca Board is used for the first async port; an auxiliary 25-pin connector connects to the Boca Board via a ribbon cable for the second async port.

## **AST I/O Mini II Card**

An AST I/O Mini II Board (ISA bus) will add one or two ports to the machine, depending on how many ports are already present. This configuration will allow you to connect two terminals to your system. This is an unintelligent, low-performance async board.

Some sample jumper settings to use the AST I/O Mini II are presented here. Place the board on a flat surface with the chips on the top surface and the card edge connector at the bottom, the DB-9 and DB-25 connectors to the right.

All of the necessary jumpers are contained in the long strip of jumpers near the bottom of the card marked E8-E17.

- The parallel port jumpers are jumpers E10 and E11. Jumper E10 enables the parallel port as LPT1; jumper E11 enables the parallel port as LPT2. Leave the jumpers off E10 and E11 if you do not want a parallel port.
- To configure the first async port as COM1 and the second as COM2, put jumpers on E8, E9, E15, and E16. Remove any jumper on E14. Use this configuration if no async ports are built into the computer.
- To configure the first async port as COM2 and not use the second async port, put jumpers on E9 and E15. Remove any jumpers on E8, E15, and E16. Use this configuration if one async port is already installed in the computer as COM1.

The 9-pin connector on the AST I/O Mini II is used for the first async port; an auxiliary 25-pin connector connects to the device via a ribbon cable for the second async port.

## INTELLIGENT ASYNC CARD SETUP

This section gives setup information for several selected intelligent multiport async adapters. This is not a complete list, but any adapter that provides a MS OS/2 device driver should work. Some manufacturers may have special drivers available specifically for MS OS/2 *MULTIUSER*. Refer to the manufacturer-supplied literature for more detail.

### DigiBoard PC/Xi

A DigiBoard PC/Xi (ISA bus) multi-port adapter allows the attachment of a large number of terminals to your system (up to 16). Several DigiBoards can be used to add more ports. Two DigiBoard PC/Xi models support 8 or 16 ports. The installation procedures are identical for both boards.



The switch settings for a Digiboard are as follows (as read along the top of the card once it is plugged into your machine):

- DS2 (8-position switch towards the rear of the machine): Switch 6 DOWN (toward PC board), all others UP.
- DS1 (11-position switch toward the front of the machine): Switches 4,6,7,8,9 UP; all others DOWN.

You will need to copy the drivers for the Digiboard to your root directory or another selected directory (for example, C:\DIGIBORD). The file names are *XA232.FCM* and *XA\_OS2.SYS*, and the files are found on the DigiBoard MS OS/2 drivers diskette. You also need to add the following configuration command as a single line in the *CONFIG.SYS* file:

```
DEVICE=<path>\XA_OS2.SYS /N:3 /B:9600 /D:N81  
/F:<path>\XA232.FCM /I:11 /P:300 /M:E80000
```

<path> refers to the path containing the files; for example, C: or C:\DIGIBORD.

## DigiBoard MC/Xi

A DigiBoard MC/Xi (Micro Channel bus) multi-port adapter allows the attachment of a large number of terminals to your system (up to 16). Three DigiBoard MC/Xi models support 4, 8, or 16 ports. The installation procedures are identical for the three boards. Several DigiBoards can be used to add more ports. There are no switches to set since this is a Micro Channel adapter.

**NOTE:** The *XA\_OS2.SYS* device driver must be at least version 1.4.3 or higher for proper operation.

The steps required to install the DigiBoard MC/Xi are:

1. Copy all of the ADF files from the DigiBoard diskette to the reference disk for your system.
2. Install the DigiBoard MC/Xi board in to your system.
3. Boot from the reference diskette and run the auto configuration program.
4. Copy the files *XA\_OS2.SYS* and *PS232.FCM* to the root directory or another selected directory (for example, C:\DIGIBORD).
5. Add the following configuration command as a single line in the *CONFIG.SYS* file, preferably before the "DEVICE=C:\OS2\COM02.SYS" line:

```
DEVICE=<path>\XA_OS2.SYS /H:M /B:9600 /D:N81  
/N:4 /F:<path>\PS232.FCM
```

<path> refers to the path containing the files; for example, C: or C:\DIGIBORD.



# APPENDIX D

## SETTING UP SPECIFIC TERMINAL TYPES

This Appendix describes general considerations for setting up a terminal for the MS OS/2 *MULTIUSER* system and provides specific terminal settings for some selected terminals.

### SOME GENERAL CONSIDERATIONS

Terminals, when they arrive from the factory, are set up with certain default settings. Your installation will probably not use the terminal in exactly the same way as it is set up at the factory, so you will have to setup the terminal according to these specific requirements.

Some of these requirements (such as keyboard character set and terminal emulation) are fixed requirements due to the nature of the terminal and MS OS/2 *MULTIUSER*; some requirements (such as baud rate) are determined by the configuration of the ports on your system and some parameters (such as keyboard key click on/off) are entirely up to you.

In the sections below, required parameters are stated with no alternate choice. Parameters marked "N/A" are not applicable to MS OS/2 *MULTIUSER* and can be left unchanged. Any settings marked "depends on configuration," such as baud rate, must be determined by the System Administrator at installation time. Note that the baud rate setting and other settings must agree with the baud rate setting and other settings specified by the CONFIG TERMINAL utility for that port or the terminal will not function correctly.



Any settings marked "up to user" can be set to the terminal user's preferences. "Key click" will cause the keyboard to emit a clicking noise when keys are pressed. "Screen saver" is a feature that will cause the screen to go blank when there is no terminal activity for a certain amount of time (usually 30 minutes, but some terminals support several different timeout intervals). "Bell volume" allows you to set the volume of the bell tone emitted by the terminal.

Most terminals have brightness and contrast controls. These are usually located on the front or the side of the terminal. These controls can be adjusted for comfortable viewing. The terminal power switch is also usually located on the front or side of the terminal. Most terminals will beep when turned on to indicate that they have started up properly. If a terminal ever fails to come up properly or displays garbled data on the screen after power-up, contact your System Administrator.

Every terminal has an associated *termtype*. This is the name used by the system to refer to the terminal type, for example, by the *CHANGE TERMINAL* utility. The *termtypes* for the various terminals are listed below.

The following section describes how to set up the various supported terminal types for use with MS OS/2 *MULTIUSER*. Please note that if a terminal refuses to go into setup mode while connected to the MS OS/2 *MULTIUSER* machine, disconnecting the terminal cable will usually allow it to enter into setup mode. Be sure to reconnect the cable after the setup process is complete.

The sections that follow give specific setup instructions for the various terminals.

**NOTE:** Since manufacturers are continually revising and upgrading their terminals, the terminal setup screens may not totally agree with the information presented in this appendix. When in doubt, refer to the terminal manual.

## ADDS 2025 Terminal

The ADDS 2025 terminal can run at up to 38,400 baud and can display 25 lines and 80 columns. To enter setup mode, press **SHIFT+SysRq**. Use the **ENTER** key to scroll through the setup screens. On a given setup screen, the **UP** and **DOWN** arrows let you select the parameter to set, and the **LEFT** and **RIGHT** arrows and the **SPACE** key allow you to choose one of the values for the parameter. Once the parameters are set, go to the EXEC1 screen, move the cursor down to the "SAVE PARAMETERS" option, and press the **SPACE** key to save the revised configuration (you should see the word DONE appear). When you are done, press **SHIFT+SysRq** again to exit setup mode. Please note that only the PC+ keyboard is supported for the ADDS 2025. The termtype for the ADDS 2025 terminal is ADDS2025.

The power switch for the ADDS 2025 terminal is on the lower right side of the video element near the back. Sliding controls for contrast and brightness are on the right side of the video element to the front of the power switch. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked SES1-EIA.

Please note that the **CTRL+Break** key does not work on the ADDS 2025 and must be redefined to another sequence.

The recommended values are:

**1. GLOBAL screen:**

- a) Screen timeout: up to user
- b) Fore/Back: Wht/Blk
- c) Scroll: Jump
- d) 44 Row Refresh: 60Hz
- e) Lines/Sessions: 26/1
- f) Printer Use: N/A

**2. COMM screen:**

- a) Mode: FDX/Line
- b) Baud Rate: depends on configuration
- c) Parity: None
- d) Parity Check: No
- e) Data Bits: 8
- f) Stop Bits: 1
- g) XON/XOFF: DC1/DC3
- h) Pace: N/A
- i) Terminator: N/A
- j) XPC: On

**3. AUX screen:**

Not applicable.

**4. KEYBOARD1 screen:**

- a) Case Select: Lower/Upper
- b) Space Char: Destructive
- c) Keyclick: up to user
- d) Keyboard: U.S.

**5. SCREEN1 screen:**

- a) Auto Wrap: Yes
- b) Auto Scroll: Yes
- c) Margin Bell: No
- d) Column Change: Erase Screen
- e) Columns: 80
- f) Cursor Home: Upper Left

6. **VISUAL1 screen:**
  - a) Cursor: Underline
  - b) Cursor Blink: Yes
  - c) Prot Reverse: N/A
  - d) Prot Half: N/A
  - e) Prot Blink: N/A
  - f) Prot Underline: N/A
  - g) Prot Suppress: N/A
  - h) Status Line: Off
7. **MODE1 screen:**
  - a) Terminal: PCTerm
  - b) Mode: Enhanced
  - c) Program Keys: Terminal Dependent
  - d) Kybd(s) to Use: N/A
8. **TAB1 screen:**

Not applicable.
9. **FUNCT1 screen:**

Not applicable.
10. **BELL:**

Use to set desired bell volume.
11. **OPTION:**

Not applicable.

## Esprit OPUS 3n1+ Terminal

The Esprit OPUS 3n1+ terminal can run at up to 38,400 baud and display 25 lines and 80 columns. To enter setup mode, press **SHIFT+SetUp**. Use the **NEXT SCRn** key to page through the setup screens. On a given setup screen, the **DIRECTION** keys let you select the parameter to set and the numeric keypad **ENTER** key allows you to scroll through the



values for the parameter. Once the parameters have been set, press the **SETUP** key once to get to the SAVE/RECALL MENU. Move the cursor to "Save" and press the **ENTER** key to save the parameters. Press the **SETUP** key again to exit setup mode. The termtype for the Esprit OPUS 3n1+ terminal is ESP3N1.

The power switch for the Esprit OPUS 3n1+ terminal is on the front of the video element on the right. A brightness control is hidden behind a door on the front of the video element to the left. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN PORT. Note that two configurations (A and B) are shown; set the A configuration values. The recommended values are:

**1. I/O Menu:**

- a) Baud Rate: depends on configuration
- b) Bits/Parity: 8N
- c) Stop Bits: 1 stop
- d) Limited Xmit

**2. Emulation Menu:**

- a) PCTerm:Scan
- b) EOL: None
- c) EOM: CR
- d) Function Keys: Default

**3. Mode Menu:**

- a) Online
- b) Full/Half Duplex: FD
- c) Monitor Off
- d) AutoWrap
- e) AutoLF/Newline
- f) Scroll
- g) No AutoPage

4. **Display Menu:**
  - a) 80 col
  - b) 24 (25) line/page
  - c) Norm Video
  - d) Jump Scroll
  - e) Std PCTerm char set
5. **Attribute Menu:**  
Not Applicable.
6. **Keyboard Menu:**
  - a) Duplex Edit
  - b) Keyclick: up to user
  - c) Repeat speed: up to user
  - d) CapsLock
  - e) RET/ENT=CR/CR
  - f) NumPad
  - g) AppCur
7. **TAB Menu:**  
Not applicable.
8. **Protocol Menu:**
  - a) RTS High
  - b) DTR High
  - c) XOFF@220
  - d) Protocol
  - e) No Host XOFF
  - f) Tools DTR High
9. **Printer Menu:**  
Not Applicable.
10. **Program Keys Menu:**  
Not Applicable.

**11. General Menu:**

- a) Top: Blank
- b) Bot: Blank
- c) Screen Saver: up to user
- d) Cursor A: On
- e) Slow Blink-Unl

**IBM 3151 Terminal (Models 51/61)**

The IBM 3151 terminal can run at up to 38,400 baud and can display 25 lines and 80 columns. Please note that only IBM 3151 models 51 and 61 are supported. To enter setup mode, press **CTRL+Setup** (keypad '-' key). Use the **ENTER** key to scroll through the setup screens. On a given setup screen, the **DIRECTION** keys let you select the parameter to set and the **SPACE** key allows you to scroll through the values for the parameter. Once the parameters are set, go to the **FUNCTION** screen, move the cursor to the **SAVE** option, and press the **SPACE** key (the word "Completed" should appear and blink). When you are done, press **CTRL+Setup** again to exit setup mode. The termtype for the IBM 3151 terminal is **IBM3151**.

The power switch for the IBM 3151 terminal is on the front of the video element on the right. A brightness control is on the front of the video element to the left of the power switch. Plug the cable to the **MS OS/2 MULTIUSER** system into the rightmost connector on the back of the video element.

The recommended values are:

**1. GENERAL screen:**

- a) Machine Mode: IBM 3151 PC or IBM 3151 PC II  
**NOTE:** IBM 3151 PC mode uses code page 437; IBM 3151 PC II mode uses code page 850. Choose whichever is more suitable for your applications.

- b) Screen: NORMAL
- c) Row and Column: 25 x 80
- d) Scroll: JUMP
- e) Auto LF: ON
- f) CRT Saver: up to user
- g) Line Wrap: ON
- h) Message Type: NON-DISPLAY
- i) Forcing Insert: OFF
- j) Tab: FIELD
- k) Term.ID: N/A
- l) Alarm Volume Level: up to you
- m) Cursor: BLINK-BAR

**2. COMMUNICATION screen:**

- a) Operating Mode: N/A (cannot be changed)
- b) Line Speed (bps): depends on configuration
- c) Word Length (bits): 8
- d) Parity: NO
- e) Stop Bit: 1
- f) Turnaround Character: N/A
- g) Interface: RS-232C
- h) Line Control: IPRTS
- i) Break Signal (ms): N/A
- j) Send Null Suppress: ON
- k) Pacing: XON/XOFF

**3. KEYBOARD/PRINTER screen:**

- a) KEYBOARD: Generated Code Set: SCAN
- b) PRINTER: Not applicable.



## Kimtron KT-70PC Terminal

The Kimtron KT-70PC terminal can run at up to 19,200 baud and can display 25 lines and 80 columns. To enter setup mode, press **ALT+CAPS LOCK**. The setup information will appear on the bottom line of the terminal. Use the **UP** and **DOWN** arrows on the numeric keypad to change a parameter value and the **LEFT** and **RIGHT** arrow keys on the numeric keypad to choose a parameter to change. Two setup lines are scrolled through with **ALT+CAPS LOCK**. The termtype for the Kimtron KT-70PC terminal is KT70.

The power switch for the KIMTRON KT-70PC terminal is on the base of the video element on the right. A brightness control is on the lower left front corner of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the main port (J1) connector on the back of the video element.

The recommended values are:

1. **First line (left to right):**
  - a) Baud rate: depends on configuration
  - b) Main Port Word Length: 8BT
  - c) Main Port Stop Bits: 1ST
  - d) Main Port Parity: NO
  - e) AUX Port Baud Rate: N/A
  - f) AUX Port Word Length: N/A
  - g) AUX Port Stop Bits: N/A
  - h) AUX Port Parity: N/A
  - i) DTR: should be blank
  - j) X-on/X-off: XPC
  - k) Refresh Rate: 60HZ
  - l) Contrast: Use **UP** and **DOWN** arrow keys to set the desired contrast level.

2. **Second line (left to right):**
  - a) Emulation: PC
  - b) Hidden/Imbedded Attribute: HIDDEN
  - c) Line/Page Attribute: PAGE
  - d) Edit Mode: DUP-EDIT
  - e) Auto New Line: AUTO-NL
  - f) Character Set: ENG
  - g) Screen Saver: up to user
  - h) Ctrl/T Function: N/A

## Link MC2PC Terminal

The Link MC2PC terminal can run at up to 19,200 baud and can display 25 lines and 80 columns. To enter setup mode, press **SHIFT+Select**. Use the function keys to select a setup screen. On a given setup screen, the **DIRECTION** keys let you select the parameter to set and the **SPACE** key allows you to scroll through the values for the parameter. Once the parameters are set, press **F12** to get to the exit screen. Press the **SPACEBAR** once to save the parameters, then press **Select** again to exit setup mode. The termtype for the LINK MC2PC terminal is LINKMC2P.

The power switch for the Link MC2PC terminal is on the right side of the video element near the front. The brightness and contrast controls are located on the underside of the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

The recommended values are:

1. **F1 (DISP) screen:**
  - a) Cursor = Blink Line
  - b) Auto Scroll = On
  - c) Func Key Line = Off

- d) Screen Saver = up to user
- e) Auto Newline = On
- f) Status Line = Off
- g) Scroll Speed = Jump
- h) Display = Normal
- i) Attributes = Page

**2. F2 (GENRL) screen:**

- a) Emulation = PCTerm
- b) Margin Bell = Off
- c) Bell Volume = up to user
- d) WPRT Intensity = N/A
- e) WPRT Rev = N/A
- f) WPRT Undrln = N/A
- g) Blk End = N/A
- h) Received CR=CRLF
- i) Monitor Mode = Off

**3. F3 screen:**

- a) Keyclick = up to user
- b) Language = US
- c) Repeat = On
- d) Keycode = Scan
- e) Enter = N/A
- f) Break = N/A

**4. F4 (COMM) screen:**

- a) Baud Rate = depends on configuration
- b) Data/Parity = 8/None
- c) Stop Bits = 1
- d) Rcv Handshake = XPC
- e) Xmt Handshake = None
- f) Comm = FDX
- g) Xmt Limit = None

**5. F5 (FKEYS):**

Not applicable.

## Link MC5 Terminal

The Link MC5 terminal can run at up to 38,400 baud and can display 25 lines and 80 columns. To enter setup mode, press **SHIFT+Select**. Use the function keys or the **Page Up/Page Down** keys to select a setup screen. On a given setup screen, the **UP** and **DOWN** arrow keys let you select the parameter to set and the **LEFT** and **RIGHT** arrow keys allow you to select the value for the parameter. Once the parameters are set, press **F1** to get back to the GEN screen, press the **S** key once to save the parameters, then press **F9** to exit setup mode. Please note that the message "Host is on Main Port" should appear on the screen. If this message reads "Host is on Aux Port," press the **P** key once to toggle the message, and then press the **S** key once to save the revision. Note that only the EPC style keyboard is supported. The termtype for the LINK MC5 terminal is LINKMC5.

The power switch for the Link MC5 terminal is on the right side of the video element near the front. The brightness and contrast controls are located on the underside of the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

The recommended values are:

1. **F1 (GEN) screen:**
  - a) Emulation: PC Term
  - b) Enhancements: On
  - c) Virtual Terminal: Off
  - d) Scroll Style: Jump
  - e) Auto Scroll: On
  - f) Auto Wrap: On
  - g) Received CR: CR



- h) AutoPage: Off
- i) Warning Bell: On
- j) Margin Bell: Off
- k) Bell Sound: up to user
- l) Block Terminator: US/CR
- m) Monitor Mode: Off

**2. F2 (COMM) screen:**

- a) Main Baud: depends on configuration
- b) Main Data/Parity: 8/None
- c) Main Stop Bits: 1
- d) Ignore 8th Bit: Off
- e) Main Rcv Hndsk: XPC
- f) Main Xmt Hndsk: None
- g) Comm Mode: Full Duplex
- h) Disconnect: 2 Sec
- i) Aux Baud: N/A
- j) Aux Data/Parity: N/A
- k) Aux Stop Bits: N/A
- l) Aux Rcv Hndsk: N/A
- m) Aux Xmt Hndsk: N/A
- n) Aux Port: N/A
- o) Aux Interface: N/A
- p) Printer: N/A

**3. F3 (DISP) screen:**

- a) Columns: 80
- b) 80/132 clear: On
- c) Lines: 25
- d) Pages: 1 x Lines
- e) Status Line: Off
- f) Cursor Style: Blink Line
- g) Cursor: On
- h) Screen Saver: up to user
- i) Background: Dark

- j) Attributes: Char
  - k) Wprt Intensity: Dim
  - l) Wprt Reverse: Off
  - m) Wprt Underline: Off
  - n) Refresh Rate: 60 Hz
  - o) Pound Char: US
  - p) Auto Font Load: On
4. **F4 (KBD) screen:**
- a) Key Clock: up to user
  - b) Key Repeat: On
  - c) Key Lock: Reverse
  - d) Return Key: CR
  - e) Enter Key: CR
  - f) Back Space Key: BS/DEL
  - g) Left Alt Key: Funct
  - h) Break: 250 ms
  - i) Xmt Limit: None
  - j) FKey Xmt Limit: None
  - k) Key Code: Scan
  - l) Language: US (or applicable national language)
5. **F5 (ANSI) screen:**  
Not applicable.
6. **F6 (FKEYS) screen:**  
Not applicable.
7. **F7 (TABS) screen:**  
Not applicable.
8. **F8 (ANSBK) screen:**  
Not applicable.

## Televideo 965 Terminal

The Televideo 965 terminal can run at up to 38,400 baud and can display 25 or 43 lines and 80 columns. To enter setup mode, press **SHIFT+Select**. Use the **LEFT** and **RIGHT** arrow keys to select the appropriate top-line menu setup screen. On a given setup screen, the **UP** and **DOWN** arrows let you select the parameter to set and the **SPACE** and **BACKSPACE** keys scroll through the values for the parameter. Once the parameters are set, go to the **COMMAND** screen, move the cursor down to **SAVE PARAMETERS**, and press the **SPACEBAR** key to save the revised values. Use **SHIFT+Select** or move the cursor to **EXIT** and press the **SPACEBAR** to exit setup mode. Note that only the Enhanced PC Keyboard is supported. The termtype for the Televideo 965 terminal is **TVI965**.

The power switch for the Televideo 965 terminal is on the front of the video element on the right. A sliding brightness control is immediately to the left of the power switch on the front of the video element. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked **MAIN** on the back of the video element.

**NOTE:** The **CTRL+Break** key does not work on the Televideo 965 and must be redefined to another sequence.

The recommended values are:

1. **GENERAL screen:**
  - a) Personality = PC-TERM
  - b) Enhance = On
  - c) Comm Mode = Full Duplx
  - d) Main/Aux = Host/Print
  - e) Monitor Mode = Off
  - f) Line Wrap = On

- g) Edit Mode = Line
- h) Font Load = On
- i) Send Acknowledge = Off
- j) Receive CR = CR

**2. DISPLAY screen:**

- a) Columns = 80
- b) Lines = 25
- c) Page Length = 1 x Lines
- d) Status Line = N/A
- e) Top Line = None
- f) Bottom Line = None
- g) Cursor Type = Undl Blink
- h) Background = Dark
- i) Screen Saver = up to user
- j) Scroll Speed = Jump

**3. KEYBOARD screen:**

- a) Keyclick = up to user
- b) Keyrepeat = On
- c) Key Lock = N/A
- d) Margin Bell = Off
- e) Return Key = N/A
- f) Enter Key = N/A
- g) Language = US
- h) Break Key = N/A
- i) Wordstar Mode = Off

**4. MAIN screen:**

- a) Baud Rate = depends on configuration
- b) Data Bit = 8
- c) Stop Bit = 1
- d) Parity = None
- e) Rec Handshake = XON/XOFF
- f) Xmt Handshake = NONE



- g) Buffer Threshold = 16
- h) Xmit Delay = None
- i) Parity Check = Off
- j) Data = 8

5. **AUX screen:**  
Not applicable.

6. **ATTRIBUTE screen:**

- a) Attribute = Char
- b) WPRT Intensity = N/A
- c) WPRT Reverse = N/A
- d) WPRT Underline = N/A
- e) 955 Attribute = N/A

7. **PROGRAM screen:**  
Not applicable.

## Wyse 60 Terminal

The Wyse 60 terminal can run at up to 38,400 baud and can display 25 or 43 lines and 80 columns. To enter setup mode, press **SHIFT+Select**. Use the function keys to select a setup screen. On a given setup screen, the **UP** and **DOWN** arrows let you select the parameter to set and the **SPACE** key allows you to scroll through the values for the parameter. Once the parameters are set, go to the **EXIT (F10)** screen, move the cursor to the **SAVE ALL** legend, and press **F10** to save the revised configuration and exit. The termtype for the Wyse 60 terminal is **WYSE60**.

**NOTE:** Only the Enhanced PC-Style keyboard is supported.

The power switch for the Wyse 60 terminal is on the right of the video element near the front. A sliding brightness control is on the front of the video element on the right. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MODEM on the back of the video element.

The recommended values are:

**1. F1 (DISP) screen:**

- a) Columns = 80
- b) Lines = 25
- c) Page Length = 1 \* Lines
- d) Attribute = Char
- e) Status Line = Off
- f) Scrl Speed = Jump
- g) Cursor = Blink Line
- h) Answerback Mode = Off
- i) Background = Dark
- j) Scrn Saver = up to user
- k) Display Cursor = On
- l) Answerback COnceal = N/A

**2. F2 (GENERL) screen:**

- a) Personality = PCTerm
- b) Comm Mode = Full Duplex
- c) Data/Printer = Modem/Aux
- d) Rcvd CR = CR
- e) Enhance = On
- f) End of Line Wrap = On
- g) Auto Scrl = On
- h) Monitor = Off
- i) Font Load = On
- j) Send ACK = N/A
- k) Init Tabs = N/A
- l) Width Change Clear = On

**3. F3 (KEYBRD) screen:**

- a) Keyclick = up to user
- b) Return = N/A
- c) Xmt Limit = None
- d) Wyseword = Off
- e) Keylock = N/A
- f) Enter = N/A
- g) Fkey Xmt Limit = N/A
- h) Language = US
- i) Key Repeat = On
- j) Corner Key = N/A
- k) Break = N/A
- l) Margin Bell = Off

**4. F4 (COMM) screen:**

- a) Baud Rate = depends on configuration
- b) Rcv Handshake = XON/XOFF
- c) Aux Baud Rate = N/A
- d) Aux Rcv Handshake = N/A
- e) Data/Stop Bits = 8/1
- f) Xmt Handshake = N/A
- g) Aux Data/Stop Bits = N/A
- h) Aux Xmt Handshake = N/A
- i) Parity = None
- j) XPC Handshake = On
- k) AUX Parity = N/A

**5. F5 (MISC):**

- a) Blk End = N/A
- b) Auto Page = Off
- c) TVI955 Attribute = N/A
- d) VP60 Blk End = N/A
- e) Labels = Off
- f) Save Labels = Off

- g) Page Edit = Off
  - h) WPRT Intensity = Dim
  - i) WPRT Reverse = Off
  - j) WPRT Underline = Off
  - k) Test = Off
6. **F6 (TABS):**  
Not applicable.
  7. **F7 (A/BACK):**  
Not applicable.
  8. **F8 (F/KEYS):**  
Not applicable.
  9. **9 (LABELS):**  
Not applicable.

## Wyse 150 Terminal

The Wyse 150 terminal can run at up to 38,400 baud and can display 25 lines and 80 columns. To enter setup mode, press **CTRL+Select**. Use the function keys to select a setup screen. On a given setup screen, the **UP** and **DOWN** arrows let you select the parameter to set and the **SPACEBAR** key allows you to scroll through the values for the parameter. Once the parameters are set, go to the **EXIT (F12)** screen and press the **SPACEBAR** key to save the revised configuration (the word **DONE** should appear). When you are done, press **F12** again to exit setup mode. The termtype for the Wyse 150 terminal is **WYSE150**.

**NOTE:** Only the enhanced PC style keyboard is supported.



The power switch for the Wyse 150 terminal is on the right of the video element near the back. Rotating brightness controls are on the right of the video element to the front of the power switch. Plug the cable to the MS OS/2 *MULTIUSER* system into the connector marked MAIN on the back of the video element.

The recommended values are:

**1. F1 (Disp):**

- a) Columns = 80
- b) Lines = 25
- c) Page = 1 x Lines
- d) Cursor = Blink Line
- e) Display = Dark
- f) Autopage = Off
- g) Scrn Saver = up to user
- h) Char Cell = 10 x 16
- i) 80/132 Clr = On

**2. F2 (Genrl):**

- a) Personality = PC TERM
- b) Scrl = Jump
- c) Rcv CR = CR
- d) Enhance = On
- e) Autoscr1 = On
- f) Monitor = Off
- g) Status Line = Off
- h) Wrap EOL = On
- i) Recognize DEL = Off

**3. F3 (Keybd):**

- a) Keyclick = up to user
- b) Xmt Lim = None
- c) Language = ENGLISH
- d) Repeat = On

- e) Margin Bell = Off
- f) Keys = Data Processing
- g) Keycode = Scan
- h) Bell Volume = up to user

**4. F4 (Comm):**

- a) Baud Rate = depends on configuration
- b) Rcv Handshake = XON-XOFF/XPC
- c) Comm = FDX
- d) Data/Parity = 8/None
- e) Rev Hndsk Level = 192
- f) Stop Bits = 1
- g) Xmt Hndshk = None

**5. F5 (Misc):**

- a) WPRT Intensity = N/A
- b) Blk End = N/A
- c) WPRT Rev = N/A
- d) Attribute = Char
- e) WPRT Undrln = N/A

**6. F6 (Tabs):**

Not applicable.

**7. F7 (Ansbk):**

Not applicable.

**8. F8 (Fkeys):**

Not applicable.



# APPENDIX E

## CABLE TYPES AND CONFIGURATION IMPLICATIONS

This Appendix contains information for wiring devices to serial data ports. The information contained here is intended to be a guide for wiring techniques to give best results in configuration, performance, and system diagnostics. Serial data ports have a variety of connectors and may be wired a variety of ways. Diagrams in this Appendix address the standard connector types. To minimize cabling errors always refer to the manufacturer's installation guide to verify the correct connector type and pin definitions. Refer to the Chapter 6 section "Host to Serial Terminal Communications."

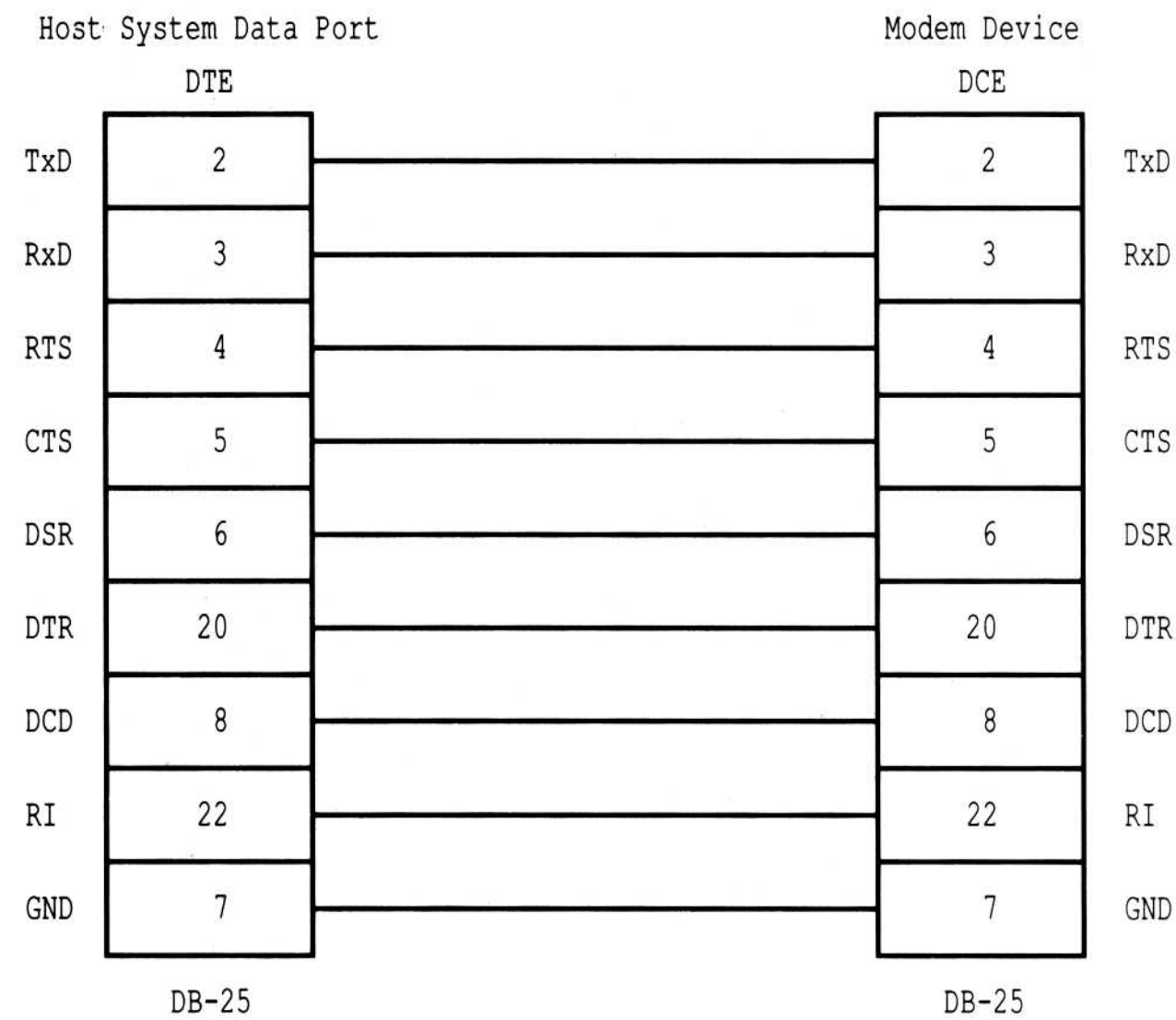
Cable wiring diagrams are provided for connecting terminals and modems to the host system. Cables wired according to the following diagrams will offer the most options for connecting a wide range of devices. The following table describes the necessary signals from the EIA RS-232 standard using 9-pin and 25-pin connectors.



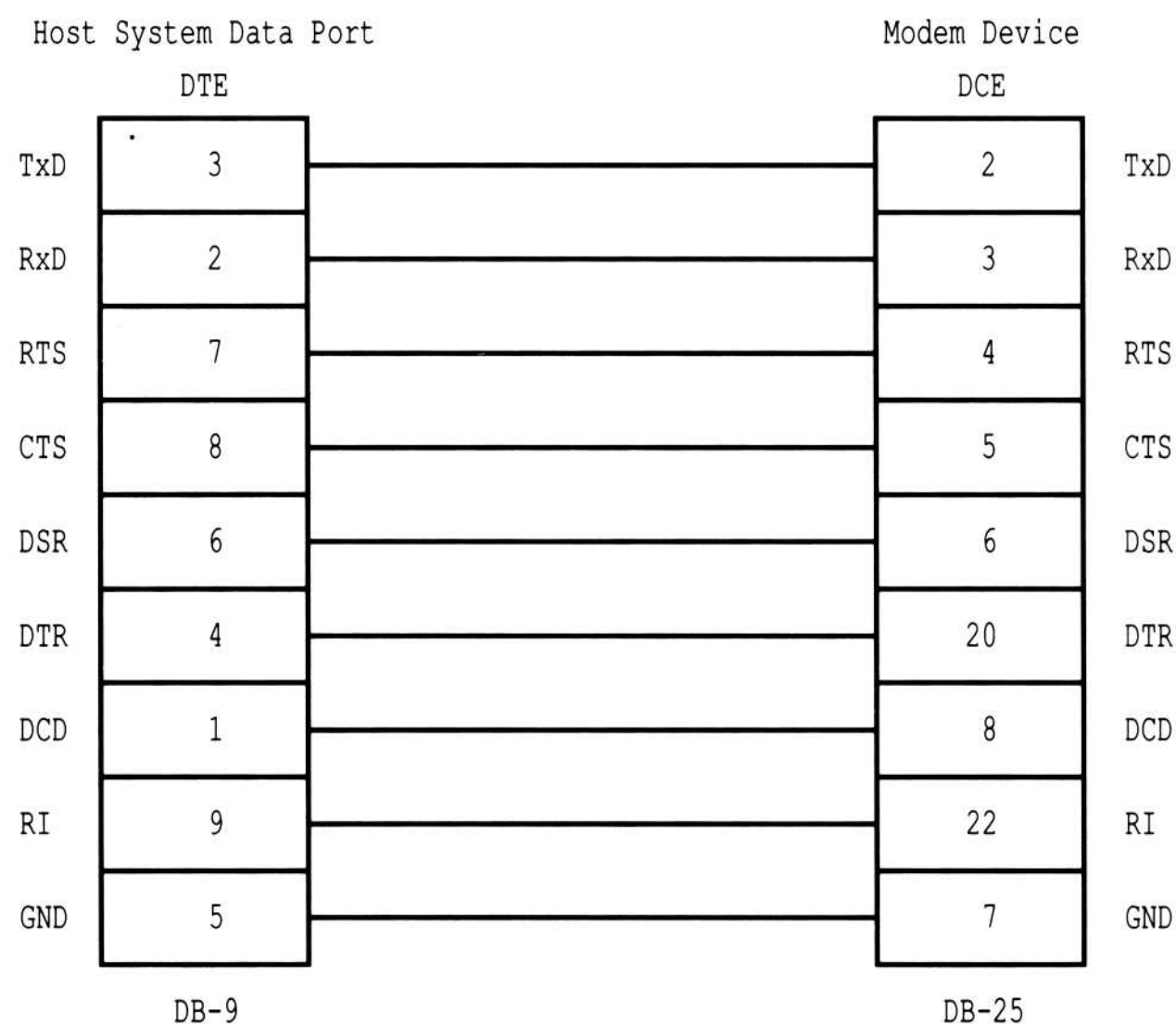
## RS-232 STANDARD DTE AND DCE PIN DEFINITIONS

DTE Connection	DB-25 Pin #	Signal Name	Description	DCE Connection
Output	2	TxD	Transmit Data	Input
Input	3	RxD	Receive Data	Output
Output	4	RTS	Request To Send	Input
Input	5	CTS	Clear To Send	Output
Input	6	DSR	Data Set Ready	Output
Ground	7	GND	Signal Ground	Ground
Input	8	CD	Data Carrier Detect	Output
Output	20	DTR	Data Terminal Ready	Input
Input	22	RI	Ring Indicator	Output

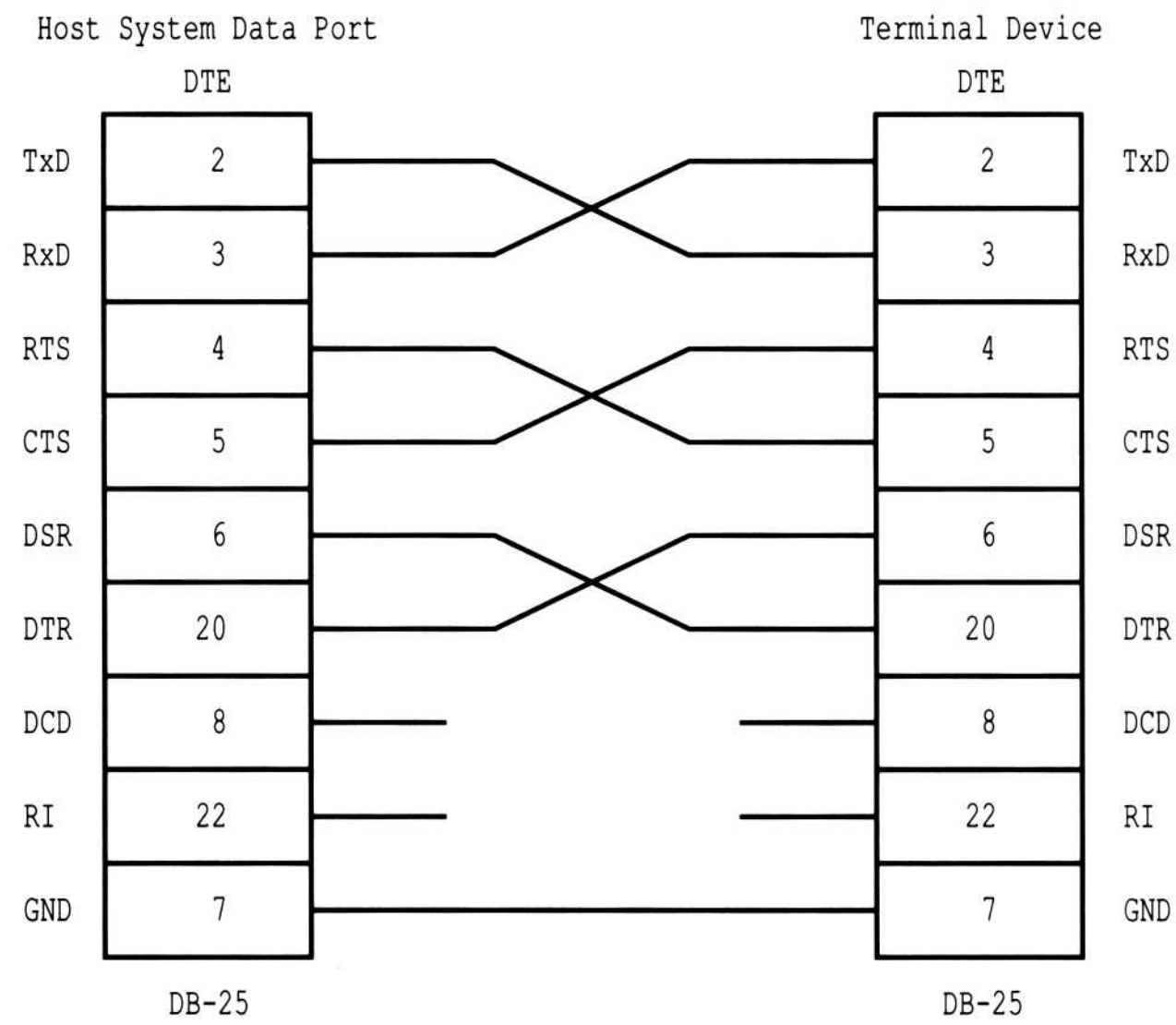
The cable connection described in the figure below is for connecting a DTE or host system data port to a DCE or modem device. This cable may be referred to as a "modem cable" or a "DCE cable".



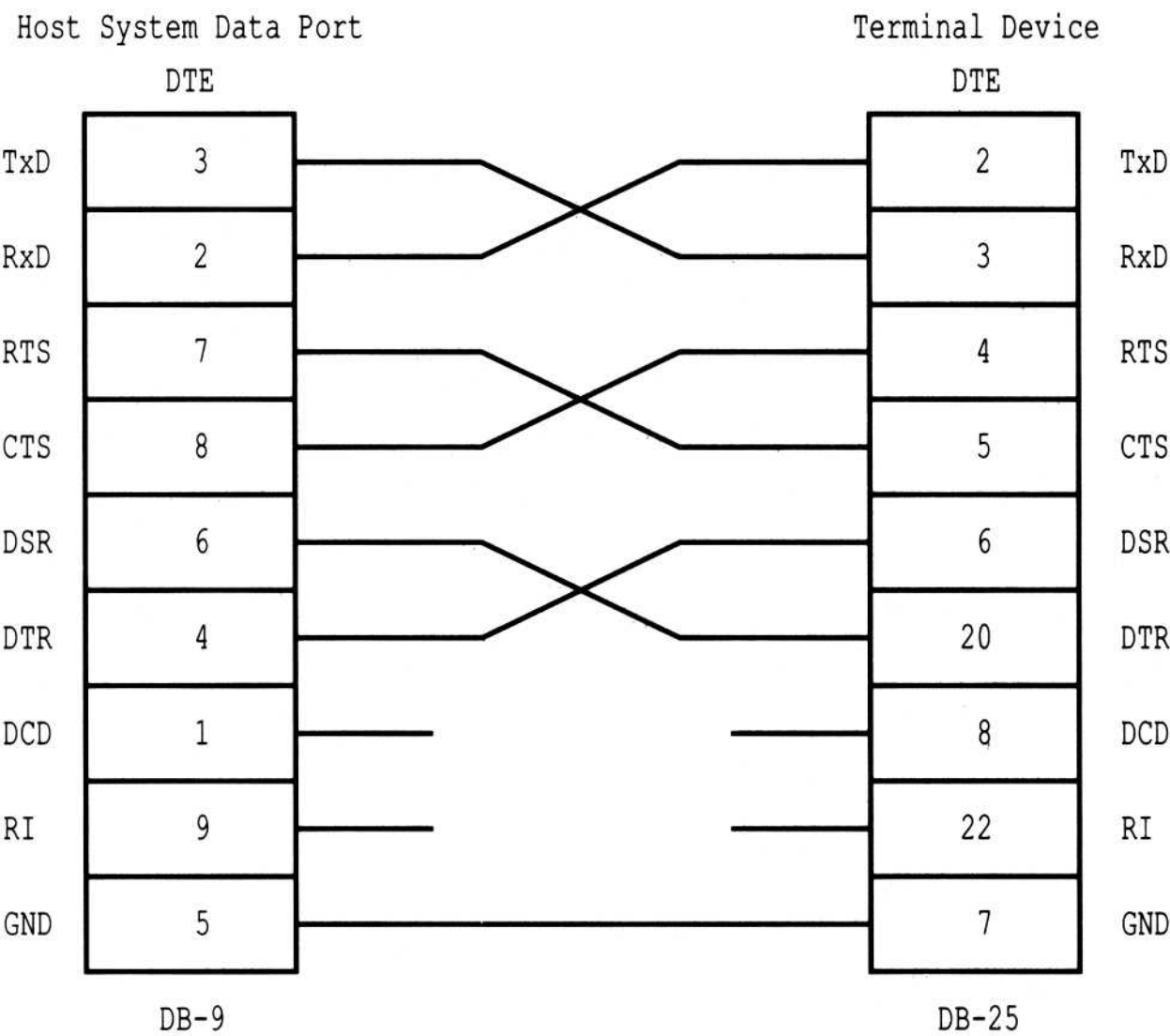
The cable connection described in the following figure is similar to the cable in the previous figure except for the DTE connector. Some systems and adapters provide a serial data port via a DB-9 connector. The pin numbers are shown for this connector under the DTE portion of the diagram. The cable is still referred to as the same name but should be specified as DB-9/DTE to DB-25/DCE. When specifying cables, the DB-XX type connector and name should always be specified.



The cable connection described in the following figure is for connecting a DTE or host system data port to a DTE or terminal device. This cable is referred to as a "Null Modem" or "DTE" cable.



The cable connection described in the following figure is similar to the cable in the previous figure except this is wired for a DB-9 connector. The pin numbers are shown for this connector under the DTE portion of the diagram. The cable is still referred to as the same name but should be specified as DB-9/DTE to DB-25/DTE. When specifying cables, the DB-XX type connector and name should always be specified.





# APPENDIX F

## SETTING UP SPECIFIC MODEM TYPES

This Appendix describes the steps required to install dial-in ports on an MS OS/2 *MULTIUSER* system. It discusses the modem hardware requirements and the software configuration requirements.

A *dial-in port* is defined as a serial port on an MS OS/2 *MULTIUSER* system associated with a modem connected to a telephone line that can be used to remotely access the MS OS/2 *MULTIUSER* system.

A *modem* is a device used to convert the digital signals from a computer or terminal into signals that can be sent over telephone lines and convert them back into digital signals at the other end. Modems also handle the details of dialing and answering the telephone. In this section, the discussion will be limited to the case where outside users call into the MS OS/2 *MULTIUSER* host computer.

System Administrators should be aware of the security implications of allowing remote dial-in to their system. Make sure all logins are secured with *passwords* to prevent unauthorized access.

## HARDWARE REQUIREMENTS

The first step is to select the serial port to be used on the MS OS/2 *MULTIUSER* system. This port can be any serial port installed on your machine. Detailed information on installing and configuring such a port is available elsewhere in this guide.

Once the port is selected, the next step is to obtain a modem and modem cable to attach to the port. The basic modem requirements are:

- The modem must be configurable so that the DCD and DSR signals are controlled by the line. When a call comes in and the connection is successfully established, those signals are raised and when the call is terminated (either normally or due to a line error), these signals are dropped.
- The modem should be switch-programmable rather than software-programmable, so the modem will remember its configuration across power failures. Software-programmable modems can be supported by manually programming them and saving the parameters into the modem's non-volatile memory. Consult the modem manual for details.
- If the modem is software-programmable, it should use the standard Hayes command set. The Hayes standard is now the de facto modem standard and it will work even if the modem is switch-programmable. Software-programmable modems that do not use the standard Hayes command set can be used but will require customization of modem control strings.

Once the modem and async port are set up, use a standard personal computer modem cable to connect the two. Refer to the section on cabling in this guide for details.

## MODEM SWITCH SETTINGS

This section gives the switch settings to use with several of the supported modems. Most other modems will be very similar to one of these.

## MultiTech 224E Modem

The MultiTech 224E modem is a 300/1200/2400 baud switch-programmable modem. The switches are found on the bottom of the modem. There is an 8-switch block and a 4-switch block. The switch settings are:

- a) 8-switch block: switches 1,3,8 DOWN; all others UP.
- b) 4-switch block: switch 4 DOWN; all others UP.

## Okidata Okitel 2400 Modem

The Okidata Okitel 2400 modem is a 300/1200/2400 baud switch-programmable modem. A block of four switches is on the back of the modem. The switch settings are: switch 2 UP; all others DOWN.

## SOFTWARE PORT CONFIGURATION

This section describes the configuration of a modem port. Use the CONFIG TERMINAL utility to configure the port.

Perform the port and driver assignments in the usual way. Choose the terminal type "Generic entry for dial-in port use" for use with this port. Set the baud rate to one of the baud rates supported by the modem. There is currently no automatic baud rate detection support, so the baud rate must be set statically at configuration time and must agree with the modem capabilities. Choose no parity and 8 data bits. Choose XPC for the flow control.

**NOTE:** Dial-in can only occur at the baud rate you configure the port for.



# GLOSSARY

## — A —

**Access Control List (ACL).** Information used by the security component of MS OS/2 *MULTIUSER* to control which secured resources like files, directories, or devices specified *usernames*, *groupnames*, or *loginnames* may access.

**Adapter Card.** An adapter card is a hardware device that is inserted into one of the expansion slots of the host computer to extend or expand the capabilities of the host computer.

**Administrator.** The most privileged security class.

**Aliasname.** See Registered Program Aliasname.

**ANSI character set.** The American National Standards Institute 8-bit character set. It contains 256 characters.

**ANSI escape sequence.** A sequence of ASCII characters used to control the keyboard and screen. The first two characters in the sequence must be the escape character and the left-bracket character.

**Application.** A program used for a particular kind of work, such as word processing or database management. See also Full-screen application.

**Archive flag.** A file characteristic that determines whether the file will be copied when you use the XCOPY, BACKUP, or RESTORE utilities. See also Flag.



**Argument.** An option and/or variable accepted by an MS OS/2 *MULTIUSER* command to clarify or enhance its meaning and use.

**ASCII character set.** The American Standard Code for Information Interchange 8-bit character set. The set consists of the first 128 (zero through 127) characters of the ANSI character set.

**Attribute.** See Extended Attribute and Security Attribute.

**Audit.** See Security Auditing.

**Autologin.** When a terminal is powered on and connected, it either displays the login prompt or it executes an autologin and runs the program that was configured to start automatically in a preconfigured *loginname* context.

— B —

**Background program.** A program that is running but cannot receive user input and does not send output to the screen. See also Foreground program and Detached session.

**Background session.** A session that is associated with a logical keyboard and display. A program running in the background session is executing but is not currently interacting with a user. It will interact with the user when it is switched to the foreground.

**Batch command.** A command used specifically in batch programs.

**Batch file.** A file that contains a batch program. Usually has a *.CMD* extension.

**Batch program.** An ASCII file that contains one or more MS OS/2 *MULTIUSER* commands. When a batch program is run, the commands are processed sequentially by CMD (in an MS OS/2 *MULTIUSER* session).

**Baud Rate.** This is the rate at which data flows across a serial cable in bits per second (bps).

**Boot disk.** See Start-up disk.

**Buffer.** A temporary storage area in the computer's memory that MS OS/2 *MULTIUSER* uses to store information.

**Byte.** A unit of information used by a computer (eight bits).

— C —

**Cable.** A cable is a wire that connects the terminal to the computer. Keyboard data sent from the terminal travels across the cable to the host computer. Video data sent from the host computer travels across the cable to the terminal.

**Cache.** See Disk cache.

**Choose.** Performs an action that carries out a command in a menu. See also Select.

**Class.** A security level assigned to a user. Class can be one of the following: Guest, User, Operator, and Administrator. Users in different classes will have permission to use different levels of user functions (like utilities) in the system.

**Cluster.** A unit of disk space that is assigned to files or directories. A lost cluster is a section of a file that is no longer associated with the original file; MS OS/2 *MULTIUSER* cannot read, write to, or modify data in a lost cluster.

**CMD.** The MS OS/2 *MULTIUSER* command interpreter. CMD translates what you type at the command prompt into instructions the computer can use. See also Command and Command interpreter.

**Code page.** A table that is used to set up language-specific versions of MS OS/2 *MULTIUSER*. You can change the code page your system uses by using the CHCP command.

**Command.** A word or phrase, usually found in a menu or typed at the command prompt, that you use to perform a task.

**Command interpreter.** A command-line interface that enables you to run commonly used "built-in" commands, utilities, batch programs, and MS OS/2 *MULTIUSER* programs.

**Command line.** The place where you type a command.

**Command-line interface.** A user interface that allows you to type commands on the command line.

**Command prompt.** The character or characters that appear on the screen at the beginning of the command line. The command prompt indicates that the computer is ready to receive input. You can customize the command prompt by using the PROMPT command.

**Communications cable.** The physical connection between a terminal and the host computer.

**Communications port.** A port that you can use to connect your computer to a terminal, a modem, another computer, or some kinds of printers. Transmits one bit at a time; also called a serial port. See also Port.

**CONFIG.SYS.** A system start-up file that contains configuration commands. See also Configuration.

**Configuration.** The way your computer is set up to operate. Configuration commands in the *CONFIG.SYS* file help you customize the way MS OS/2 *MULTIUSER* runs on your computer.

**CONFIGUSR.** A startup file that is run when a user logs in. There can be one for each *username* (home directory).

**Connected login.** Programs running in the context of a connected login interact with the user through the display and keyboard of the terminal where the login is connected.

**CPU (central processing unit).** The part of the computer that receives and carries out instructions.

**Current directory.** The directory you are working in.

**Current drive.** The drive that you are currently using.

**Cursor.** Usually a blinking line or small box on the screen that shows where the next character you type will appear.

— D —

**Data file.** A file that contains information.

**Data path.** The search path for data files.

**Default.** Describes an option, command, or device that is automatically selected or chosen by the system. You can override a default setting by selecting the appropriate option, command, or device.



**Default directory.** The directory that the system uses if a specific directory is not given to the program or command.

**Destination disk.** The disk to which you copy or move files. See also Source disk.

**Destination file.** A file to which information is copied. See also Source file.

**Detached session.** A session that contains a program that can run completely in the background and will never interact with you except through a popup session.

**Device.** A component of the system's hardware configuration, such as a modem, printer, or disk drive.

**Device driver.** A program that controls how a computer interacts with another device, such as a printer.

**Dial-in port.** A communications port with a modem attached to it that can accept communications connections originated from terminals over a phone line.

**DIRECTION keys.** The four arrow keys on the keypad. The names of the individual DIRECTION keys refer to the direction the arrow points: the **UP** key, the **DOWN** key, the **RIGHT** key, and the **LEFT** key.

**Directory.** Part of a structure for organizing your files into convenient groups. A directory is like a file drawer that holds a particular group of files. A directory can contain both files and other directories (sometimes called subdirectories). You can see the directories on your system by using the DIR command at the command prompt. See also Subdirectory.

**Directory path.** See Path.



**Disconnected login.** A login that is not attached to a terminal. Programs in a disconnected login run in the background. You may employ the DISCONN utility to disconnect the login from the terminal if you need to leave your terminal but keep the programs running.

**Disk.** A storage medium: fixed disk and floppy disk (5¼-inch and 3½-inch disk).

**Disk cache.** A buffer in which MS OS/2 *MULTIUSER* stores information that it has recently read from your fixed disk. When an application is instructed to read information from the fixed disk, it first checks for the information in the disk cache.

**Disk drive.** Hardware that is used to read from and write to a floppy disk or fixed disk.

**Diskette.** See Floppy Disk.

**Display.** See Screen.

**Drive letter.** The letter used to label a drive — for example, C. See also Disk drive.

**Dynamic-link library.** A replaceable component of a program file. Many MS OS/2 *MULTIUSER* applications use dynamic-link libraries to reduce the size of program files and to allow program files to share resources.

— E —

**Environment variable.** A string consisting of environment information, such as a drive, path, or *filename*, associated with a symbolic name that can be used by MS OS/2 *MULTIUSER*. Use the SET command to define environment variables.

**Error message.** See Message.

**Escape sequence.** Codes that affect the appearance of the screen, the movement of the cursor, and the use of the keyboard keys — for example, ANSI escape sequences.

**Executable file.** See Program file.

**Expanded memory.** MS OS/2 *MULTIUSER* does not use expanded memory. If you have expanded memory in your computer, you should convert it to extended memory. See your memory board manual for details.

**Extended attribute.** Additional information, such as author, application type, and file history, that some MS OS/2 *MULTIUSER* applications attach to files and directories. Extended attributes are used by other applications, the file system, or the operating system itself.

**Extended Industry Standard Architecture (EISA).** See Industry Standard Architecture.

**Extended memory.** MS OS/2 *MULTIUSER* uses extended memory.

**Extended partition.** An optional partition on a fixed disk that can contain logical drives. The logical drives can contain additional files. See also Partition.

**Extension.** See Filename extension.

— F —

**FAT (File Allocation Table).** A map of a disk that indicates which clusters are allocated to each file and in what order. A file system that uses this type of mapping is referred to as a FAT file system.

**File.** A collection of related information treated as a unit. There are different kinds of files, including program files and data files. For example, a file might contain the text of a letter.

**File allocation table.** See FAT.

**File extension.** See Filename extension.

**File flag.** See Flag.

**Filename.** A sequence of ASCII characters used to identify a file. In the FAT file system, a *filename* can be up to 8 characters long, with an optional extension of up to 3 characters. In the High-Performance File System (HPFS), the *filename* can be up to 254 characters long and can include characters that are not allowed in the FAT file system.

**Filename extension.** The period and up to three letters at the end of a *filename* in the FAT file system. The extension identifies the kind of information a file contains. For example, the extension *.CMD* indicates that the file contains a batch program. Some applications append an extension to the files that you create with them.

**File system.** The part of the operating system that takes care of such tasks as disk-space management, file storage, and directory structure.

**Filter.** A utility that processes input from a device or file and sends it to an output device or file. Refer to the FIND, MORE, and SORT commands.

**Fixed disk.** A disk that can store much more information than a floppy disk and is generally installed in the computer.

**Flag.** A characteristic of a file that you can turn on or off. There are four file flags: archive, hidden, read-only, and system.

**Floppy disk.** A 3½-inch or 5¼-inch disk that can easily be inserted into and taken out of your computer and that is used for storing programs and files.

**Font.** A complete set of characters of a particular design and size.

**Foreground session.** The session is associated with the physical keyboard and display on the terminal that you are working with. You interact with the program in the foreground session.

**Format.** Prepares a disk so that it can hold information. Formatting a disk divides the storage area on the disk into sections and erases whatever information was on it.

**Fragmentation.** This describes the state of physical memory where many small free blocks exist.

**Full-screen application.** An MS OS/2 *MULTIUSER* application runs in a full-screen session.

**Full-screen session.** A separate environment where MS OS/2 *MULTIUSER* applications run in a full-screen. You can start multiple full-screen sessions. From a full-screen session, you can switch to other applications by pressing **ALT+ESC** or to the Task List by pressing **CTRL+ESC**. See also Session.

**Function keys.** The keys labeled F1, F2, and so forth. The purpose of the function keys is defined by the application that uses them. For example, in some applications, pressing the F1 key displays Help information.



## — G —

**Group.** A collection of users who have similar administration configuration characteristics and/or similar security access list characteristics.

**Groupname.** The last part of the *loginname*. Sometimes descriptive of the collection of users placed in the same group.

**Guest.** The least privileged security class.

## — H —

**Hard disk.** See Fixed disk.

**Hard error.** An error condition, such as an open disk-drive door or an incorrectly formatted disk.

**Hardware.** The equipment that makes up your computer system, including such items as the keyboard, disk drives, and monitor.

**Hidden file.** A file that cannot be viewed.

**High-density floppy disk.** A 5¼-inch floppy disk formatted to store approximately 1.2 megabytes of information. A 3½-inch floppy disk formatted to store approximately 1.4 megabytes of information.

**Home directory.** A directory created by MS OS/2 *MULTIUSER* when a user is added to the system through the CONFIG USER utility. Primarily used to hold a user's startup files. This directory applies to one and only one user (username).



**Host computer.** The host computer is the computer on which MS OS/2 *MULTIUSER* is running. The host computer typically contains a fixed disk, one or more floppy diskette drives, memory, a display, and a keyboard.

**Hotkey.** A hotkey is a key stroke or combination of key strokes that when pressed, cause the operating system to perform a function, as opposed to an application program. For example, to switch between one application and another application press **ALT+ESC**. This combination of key strokes is known as a hotkey.

**HPFS (High-Performance File System).** A file system that can be installed in addition to the FAT file system. HPFS uses disk caching and lazy writing, and allows *filenames* that are longer than FAT *filenames* and that can contain characters not allowed by the FAT file system. MS OS/2 *MULTIUSER* requires use of HPFS for the primary partition of the boot drive.

— I —

**IFS.** See Installable file system.

**Industry Standard Architecture (ISA).** Personal computer adapter cards and slots compatible with the PC/AT. Other architectures include Micro Channel and Extended Industry Standard Architecture (EISA).

**Initial default directory.** See working directory.

**Installable file system (IFS).** A file system installed in addition to the FAT file system.

— K —

**Kilobyte (KB).** Approximately one thousand bytes. (The exact number is 1024 bytes.)

## — L —

**Label.** An argument of the GOTO batch command that controls which part of the batch file the batch program switches to. See also Volume label.

**Lazy writing.** Sending (writing) information from a disk cache to the disk when the disk is idle or a specified amount of time has passed.

**Licensed number of users.** A count of connected logins within a host that represents the number of users licensed for that host.

**Log file.** A file that is used to store system event manager information for problem determination.

**Logical drive.** An extended partition on a fixed disk that is treated as a separate disk drive. For example, a 60-megabyte fixed disk could be divided into logical drives C and D, each of which could have 30 megabytes of storage space. See also Virtual drive.

**Login.** As a verb: What you do in order to be able to use the MS OS/2 *MULTIUSER* system.

As a noun: The complete set of sessions and programs that you work with after completing a specific login sequence. Your identity for a given login is based on the *loginname* you used to login.

**LoginID.** An identification number for a login. This ID is a host-based number and is sufficient to specify a login within a host.

**Loginname.** Also known as "username[groupname]". It is the identity you use during the login process. The identity that MS OS/2 *MULTIUSER* has for you while logged in. If a username is in two groups, the groupname is required as part of the loginname in certain contexts.

**Login prompt.** What MS OS/2 *MULTIUSER* displays on the screen in order for you to login.

**Logout.** Terminating your interaction with the system. After logout, a user must login before interacting with the system again.

**Low-density floppy disk.** A 5¼-inch floppy disk formatted to store approximately 360K of information.

— M —

**Maximum guaranteed limits.** The maximum amount of resources that a *loginname* can allocate if resource management is enabled by the System Administrator. The System Administrator can preconfigure each *loginname* to have different maximum resource limits.

**Megabyte (MB).** Approximately one million bytes.

**Memory.** The place where data and programs are stored while your computer is turned on; often referred to as random-access memory or RAM. Any information that is stored in memory is lost when you turn off your computer. To permanently save information located in memory, you need to copy the information to a disk. Memory is often measured in kilobytes and megabytes.

**Message.** A sentence or short paragraph that may appear on your screen. An error message contains a message-identification number that consists of three letters followed by a four-digit number. You can get more information about an error message by typing help and the message number at the command prompt.

**Micro Channel.** See Industry Standard Architecture.

**Minimum guaranteed limits.** A guarantee of resource availability for each *loginname* after a login. A *loginname* will be denied the ability to login if the minimum resource limits cannot be satisfied.

**Modem.** Hardware that allows the connection between the terminal and the host computer to span long distances and utilize phone lines.

**Multitasking.** A feature of MS OS/2 *MULTIUSER* that lets you run more than one program at the same time.

**Multiuser system.** A feature of MS OS/2 *MULTIUSER* that allows multiple people to use the host computer simultaneously.

— N —

**NUL.** A destination to which you can redirect output. Anything sent to NUL is discarded.

**Numeric keypad.** The part of your keyboard that resembles a 10-key adding machine or calculator keyboard. In addition to numbers, the numeric keypad also contains the **HOME**, **END**, **PAGE UP**, **PAGE DOWN**, and **DIRECTION** keys.



— O —

**Operating system.** A collection of instructions that tell your computer how to run programs and applications. MS OS/2 *MULTIUSER* is a multitasking and multiuser operating system.

**Operator.** The next to the most privileged security class.

**Option.** A predefined argument you can use on the command line to modify the way a command operates.

**Owner.** A security attribute for a file or directory denoting the *loginname* of the user who has the rights to alter security attributes of the file or directory.

— P —

**Parent directory.** Any directory that contains other directories. See also Subdirectory and Root directory.

**Partition.** A section created on the fixed disk by the operating system to organize the contents of the disk. Different partitions can contain different file systems. See also Extended partition.

**Password.** A character string known only to you that you use to login. Do not divulge the password to anyone else.

**Path.** A collection of directory names separated by backslashes (\). A path describes the location of a directory within the directory structure of the system.



**Permissions.** Flags associated with an ACL entry denoting the entry's security access rights. Examples are Read (R), Write (W), and Delete (D).

**Pipe.** A symbol (|) used to redirect the output from one command and use it as input for the next command.

**Placeholder.** A word in a syntax line that represents specific information you will need to supply.

**Popup session.** A program running in a background session can create a temporary foreground session by creating a popup session to temporarily interact with you.

**Port.** A slot on your computer to which you can connect a terminal, printer, modem, or other input or output device. MS OS/2 *MULTIUSER* recognizes three printer ports (LPT1, LPT2, and LPT3) and at least three communications ports (COM1, COM2, and COM3).

**Primary code page.** The primary (default) character set used by your computer. See also Code page.

**Primary partition.** The partition on your fixed disk that must contain system start-up files.

**Printer queue.** Where print jobs are stored until the printer becomes available.

**Print job.** A file waiting to be printed. A contiguous set of data to be printed together.

**Print job shadow.** If a print job is in the middle of being printed and the system shuts down, the print job can be reprinted from the print job shadow when the system comes back up.

**ProcessID.** An identification number for a process. This ID is a host-based number and is sufficient to specify a process within a host.

**Processor.** The integrated circuit where your computer interprets and processes information; also called the central processing unit or CPU.

**Program.** A set of instructions that tells the computer how to perform a task. See also Application and Batch program.

**Program file.** A file that contains a program. In the FAT file system, the file must have one of the following *filename* extensions: *.CMD*, *.EXE*, or *.COM*.

**Programming language.** A notational system used to create a computer program.

**Program Selector.** A program that provides an interface to start and switch between programs in different sessions. Usually the first program to be run in your login.

**Prompt.** See Command prompt.

## — Q —

**Quiesce.** The process of bringing a device or system to an inactive state.

**Queue.** See Printer queue.

## — R —

**Read-only flag.** A file characteristic that determines whether or not the contents of a file can be modified. See also Flag.

**Redirect.** Sends the output from a command to a file instead of the screen.

**Registered Program.** A program that has been identified to the security system so that it can be given special access permissions via an ACL. See also Registered Program Aliasname.

**Registered Program Aliasname.** A name (alias) given to a program or program path to give it a specific identity for use with the security system. It is analogous to your *loginname*.

**Replaceable parameter.** A command option you can define each time you run a batch program. A replaceable parameter is represented by a percent sign (%) followed by a digit from zero through 9.

**Root directory.** The primary directory on a disk. All other directories on the disk are secondary. The root directory is represented by a backslash (\).

**Run.** Starts an application.

— S —

**Save.** Stores a file or changes to a file on a disk.

**Screen.** The surface upon which information is displayed on a video device.

**Scroll.** Move text up or down or left or right.

**Search path.** The list of directories in which MS OS/2 *MULTIUSER* searches for a file or directory.

**Secondary code page.** The alternate character set you can use instead of the primary code page. See also Code page.

**Sector.** Part of a track on a disk; the smallest unit of disk space that the computer can access. Sectors are created when the disk is formatted. See also Track.

**Security Attributes.** Information associated with a named resource describing security characteristics. Security attributes are ACLs, audit data, and ownership data.

**Security auditing.** The ability to log attempted security violations for later review.

**Separator page.** What can be printed out in order to distinguish between different print jobs.

**Serial port.** A serial port, or asynchronous communications port, is hardware that allows for transmitting and receiving data between the host computer and the terminal. Terminal subsystems contain one or several serial ports. Similarly, all host computers generally contain at least one serial port.

**Session.** A set of logical devices like a display and keyboard that a program needs to interact with you. Every application program executes in the context of a session. You switch between the different applications in your login by switching between sessions with a hotkey sequence or by using the Program Selector.

**SessionID.** An identification number for a session. This ID is a login-based number and is sufficient to specify a session within a login but not within a host.

**Shortcut key.** A special key or key sequence, available for some commands, that you can press to carry out the command without first selecting a menu. The shortcut keys for a command are often listed on the menu to the right of the command name.

**Software.** The set of instructions that make computer hardware perform tasks. Both the operating system that allows you to run applications and the applications themselves are software.



**Source disk.** The disk from which you copy or move files. See also Destination disk.

**Source file.** A file from which information is copied. See also Destination file.

**Spool.** Storing print jobs in the printer queue.

**Spooler.** A program that organizes and prioritizes print jobs.

**Standard error.** The destination of error messages sent by your computer. The computer usually sends error messages to the screen but you can redirect them to other destinations.

**Standard input.** The source of input to your computer. The computer usually receives input from the keyboard but you can redirect input from other sources.

**Standard output.** The destination of output from your computer. The computer usually sends output to the screen but you can redirect output to other destinations.

***STARTUP.CMD.*** A startup batch file. A *STARTUP.CMD* file located in the root directory of the boot drive is a system startup file and is processed when the system starts. A *STARTUP.CMD* file located in a user's working directory is a user-specific startup file and is processed by the Program Selector.

**Start-up disk.** The disk that contains your MS OS/2 *MULTIUSER* start-up files.

**Start-up drive.** The drive that contains your MS OS/2 *MULTIUSER* start-up files.

**Start-up files.** The files that set up the working environment for you or the system.



**Starved priority.** After a program has been waiting to run for some time period (as defined by the *CONFIG.SYS* command *MAXWAIT*), its priority is increased to starved priority. This feature is disabled if *PRIORITY=ABSOLUTE* in the *CONFIG.SYS* file.

**String.** Characters, terms, or other symbols in a connected sequence.

**Subdirectory.** A directory contained within another directory. All directories are subdirectories of the root directory.

**Swap.** Temporarily transfers an application or information between system memory and the fixed disk.

**Switch.** Moves from one application to another or from one session to another. You can switch between applications by using the keyboard or the Task Selector.

**System console.** The host computer's display and keyboard is considered a unique terminal, called the system console.

**System global.** An application/Dynamic Link Library (DLL) that requires its named resources, DLL initialization routines, and DLL shared data to be shared across all user contexts. These EXE's/DLL's are usually multiuser enabled with or without MS OS/2 *MULTIUSER*. The REGISTER utility is used to mark a *.EXE* or *.DLL* system global.

**System Semaphores.** This NAMED resource is used by applications/Dynamic Link Libraries (DLL) to control execution flows between multiple threads/processes. A finite number of system semaphores is available in the system (256). Resource management can be used to specify the minimum guaranteed and maximum allowed number of system semaphores per user.

---

— T —

**Task List.** A list of your current activities. The Task List can be used to switch among the applications you have running on your computer.

**Terminal.** A terminal is a combination of a computer display and keyboard. The terminal's keyboard is the primary means by which you enter commands and keyboard data. Results of those commands and display data are shown on the terminal's display. MS OS/2 *MULTIUSER* supports a variety of terminals that attach to the computer's serial port (refer to Appendix D, "Setting Up Specific Terminal Types").

**Terminalname.** The name given to the terminal by the System Administrator when the terminal was configured. QUERY TERMINAL can be used to determine the terminalname.

**Terminaltype.** The terminaltype is the type (manufacturer/model) of terminal. See Appendix D, "Setting Up Specific Terminal Types" for information on the terminal type of many of the different terminals that are supported. QUERY TERMINAL can be used to determine the terminaltype.

**Terminal subsystem.** A terminal subsystem is the hardware in the host computer that transports the data to and from the terminal. There are two types of terminal subsystems in MS OS/2 *MULTIUSER*; the system console terminal subsystem and the serial port terminal subsystem.

**Text file.** See Data file.

**Thread.**Part of a program that can be scheduled by MS OS/2 *MULTIUSER* to run on its own. A thread is a unit of execution within a process. A process can have multiple threads. The maximum number of threads in the system is 1024. Resource management can be used to specify the minimum guaranteed and maximum allowed number of threads per user.

**Time slice.** The amount of execution time the system will give a thread before reassigning the CPU to another thread.

**Track.** A path where information is stored on a disk. A track is made up of sectors. See also Sector.

— U —

**User.** The next to the least privileged security class.

**User home directory.** See Home directory.

**User interface.** The part of the operating system you interacts with. See also Command-line interface.

**User Global.** An application/Dynamic Link Library (DLL) that requires its NAMED resources, DLL initialization routines, and DLL shared data to be shared within a user login context. This is the default context assumed for every application/Dynamic Link Library (DLL). The REGISTER utility is used to make a *.EXE* or *.DLL* user global if it had been previously marked system global.

**User login.** see Login.

**User login global.** See User global.

**Username.** The first part of the *loginname*. The System Administrator decides what the username will be.

**Utility.** A program provided with MS OS/2 *MULTIUSER* that performs such system-maintenance tasks as copying files or formatting disks. For example, BACKUP is an MS OS/2 *MULTIUSER* utility for creating backup copies of your files.

— V —

**Virtual disk.** A section of RAM that is set up as a disk drive. Read and write operations are faster to a virtual disk than they are to a fixed or floppy disk. See also Memory.

**Virtual memory.** This term is used to describe both the physical memory present in the machine plus external (disk) space used to swap out memory. The amount of Virtual Memory in the system is a direct relation to the amount of physical memory and free disk space in the host CPU.

**Volume label.** The name you assign to a disk as part of the disk-formatting process.

**Volume serial number.** An individual disk-identification number assigned by MS OS/2 *MULTIUSER* as part of the disk-formatting process.



## — W —

**Wildcard character.** A character (either \* or ?) that can be included in a *filename* to indicate any character or group of characters that might match that position in other *filenames*. For example, \*.*EXE* represents every file in a directory that ends with the *.EXE filename* extension. The (?) character refers to a single character; for example, X?*Z.EXE* represents the files *XYZ.EXE* and *XDZ.EXE*, but not *XABCDZ.EXE* or *XZ.EXE*.

**Working directory.** A directory used as the initial directory when you log into the system. This may be different from a your home directory. Primarily used to contain task-oriented data or programs. This directory may be shared among several users as their working directory.

**Write-protected disk.** A floppy disk you can read from but not write to.



## INDEX

- ? (question mark) 12-11
- \* (asterisk) wildcard 12-11
- Access Control Lists (ACLs) 1-9
  - explicit versus implicit 7-34
  - viewing 7-38
- Access Name 7-33
  - changing 7-36
- Access Permissions for Programs 7-17
- ACL
  - explicit versus implicit 7-34
  - viewing 7-38
- Adapter card
  - defined 6-2
- ADD5 2025 Terminal D-3
- Administrator
  - security class 7-6
- Administrator class
  - deleted, create new user 16-7
- Alias, logical drive letter 10-7
- Aliasname
  - registered program 7-16
- Application 4-2
  - access to 11-7
  - categorize 11-3
  - installing 11-4
  - setting up users 11-6
  - work files 11-8
- ASA
  - Audit Security Attribute 7-41
- AST I/O Mini II Board C-4
- Asynchronous communications 12-56
  - Device Driver, defined 6-2
- Asynchronous port
  - multiport cards C-1
- Audit Events 7-26
  - Secondary 7-28
- Audit flags
  - system profile 5-9, 5-19
- Audit Log 7-28
- Audit Security Attribute (ASA)
  - add or change 7-41
  - indicator 7-33
  - removing 7-42
  - viewing 7-41
- Audit Targets 7-27
- Auto connect login 5-19
- Auto disconnect login 5-19
- Autologin 12-6
  - parameters 6-14
  - parameters, changing 6-32
- Boca Research IO/AT Board C-2
- Background program
  - starting 9-17
- Backing up
  - open system files 13-16
  - system directories 13-13
- BACKUP utility 13-14, 13-16
  - disk, adding files 12-16
  - system profiles 13-14
- Backup Copies
  - files, modified 12-15, 12-16
- Backup Files
  - restoring 12-18
- Backup Log
  - making 12-17
- Batch Files 4-7
- Baud Rate 12-56
  - defined 6-2
- Bold type 5
- Brackets ([ ]) 7
- Buffer
  - size, receive 6-25
  - TRACEBUF command 9-23
- BUFFERS command 9-3, 9-14
- Cable
  - defined 6-2
  - wiring diagrams E-1
- Capital letters 5
- CHANGE utilities 12-45
  - AUDIT 12-45
  - EVENTS 12-46
  - EVENTS, event selection 14-4
  - LIMITS 12-46, 16-10

- TERMINAL 12-47
- Changing
  - code pages 9-18
  - CONFIG.SYS* 9-5
  - display device driver 10-4
  - permissions,multiple files 7-37
- Character set
  - CODEPAGE command 9-5
- Characters per line
  - display mode 12-59
  - parallel printer 12-55
- CHCP command 9-18, 9-20
- CHKDSK 13-4, 13-5
  - /V option 13-9
  - virtual disk drive 10-10
- Class
  - access 7-12
  - Administrator 1-11, 7-6
  - Guest 1-11, 7-8
  - Operator 1-11, 7-7
  - User 1-11, 7-8
- Class level 5-20
- Code page
  - installation 9-18
  - printer 9-22
  - setting up 9-18
  - support,system console 9-23
  - support,terminal 9-23
- CODEPAGE command 9-3, 9-20
- Colon (:) 8
- Color display 12-59
- COM01.SYS device driver 10-6
- COM02.SYS device driver 10-6
- Command 4-2
  - notational conventions 5
- Communication port
  - configure 8-6
  - device driver 10-6
  - MODE utility 12-55
- Communications
  - host to serial terminal 6-18
- COM*n* 12-56
- Compare
  - floppy disks 12-28
- CONFIG
  - SYSTEM, event selection 14-4
  - SYSTEM, profile panel 5-5
  - utility, introduction 4-8
- CONFIG ACCESS
  - using 7-32
- CONFIG GROUP
  - Audit flags 5-11
  - Class level 5-12
  - Copying a Group 5-14
  - Creating a New Group 5-13
  - Deleting a Group 5-15
  - First user program 5-12
  - Login time restrictions 5-12
  - Modifying a Group 5-15
  - Renaming a Group 5-17
  - Resource limits 5-12
  - Working directory 5-11
- CONFIG PROFILES 5-11, 7-2
  - Adding Users to a Group 5-26
  - Changing a User's Password 5-28
  - Copying a Group 5-14
  - Copying a User 5-22
  - Creating a New Group 5-13
  - Deleting a Group 5-15
  - Deleting a User 5-23
  - Deleting a User From a Group 5-27
  - edit user profile 5-18
  - Modifying a Group 5-15
  - Modifying a User 5-24
  - Renaming a Group 5-17
  - Renaming a User 5-25
- CONFIG TERMINAL
  - configuration, refreshing 6-33
  - configuration, resetting 6-32
  - Data Flow 6-19
  - device name, changing 6-31
  - system defaults 6-31
  - terminal profile, copying 6-29
  - terminal profile, deleting 6-27
  - terminal profile, moving 6-28
  - terminal profile, renaming 6-27
  - terminal subsystem, creating a new 6-33

- terminal type, changing 6-30
- terminal type, defaults 6-32
- CONFIG USER
  - Adding Users to a Group 5-26
  - Creating a New User 5-21
  - Deleting a User 5-23
  - Deleting a User From a Group 5-27
  - edit user profile 5-18
  - Modifying a User 5-24
  - Renaming a User 5-25
- CONFIG.SYS 4-4, 9-24
  - backup copy 9-5
  - defining 9-3
  - installation 9-5
  - modifying 9-5
  - security 11-5
  - SET 10-4
- CONFIG.USR 4-4, 4-6, B-3
- Configuration
  - viewing options 7-35
- Configuration Files 4-6
- Configure
  - other drivers 7-42
  - communication port 8-6
- Configuring
  - parallel port 12-55
  - serial port 12-56
  - terminals, cables, serial ports 6-9
- Connection Type 6-23
- Continuous retry 12-57
- Copying
  - floppy disk 12-25
- Copying a File or Directory 12-10
- COUNTRY command 9-3, 9-19
- CTRL key
  - stopping a program 9-5
- CTS (clear to send) 12-58
- Data Bits 6-20, 12-57
  - defined 6-2
- Data Flow 6-19
- Data Format
  - connection type 6-23
  - data bits 6-20
  - flow control 6-21
  - handshaking techniques 6-20
  - parity 6-21
  - stop bits 6-21
- Data paths
  - access to 11-7
- DATE
  - country settings 9-19
  - setting 12-52
  - setting, displaying 12-52
- DDINSTAL utility 10-3
- Default autologin user
  - system profile 5-6
- Default first user program
  - system profile 5-6
- Default hotkeys
  - system profile 5-7
- Default resource limits
  - system profile 5-9
- Default user/terminal for system
  - startup commands
  - System profile 5-6
- DEFAULT.SEP 8-4
  - example 8-14
- DEVICE command 9-3, 10-3, 10-6
- Device driver
  - add 11-6
  - changing display 10-4
  - COM01.SYS 10-6
  - COM02.SYS 10-6
  - description 9-14, 10-2
  - EXTDSKDD.SYS 10-7
  - installation 10-2
  - setting up 9-14, 10-3
  - VDISK.SYS 10-8
- Device Name
  - changing 6-31
- DEVINFO 9-3
  - KBD 9-21
  - SCR 9-21

- Dial-In
  - terminal, configuring 6-25
- Dial-in port
  - defined F-1
- DigiBoard MC/Xi C-6
- DigiBoard PC/Xi C-5
- Directory
  - compressed 7-33
  - copying HPFS to FAT 12-10
  - expanded 7-33
  - Home, \USR\loginname 7-30
  - moving HPFS to FAT 12-10
  - structure 11-2
  - viewing files 7-34
- Disk
  - damaged, recover file 13-10
  - displaying information 13-5
  - fixing errors 13-7
  - formatting 12-21
  - restoring files 12-18
  - swap file 9-13
  - volume label 12-26
- DISKCACHE command 9-11
- DISKCOMP utility 12-28
- DISKCOPY utility 12-25
- DISPLAY.DLL file 10-5
  - device driver 10-4
- Display, device driver 10-4
- Displaying
  - volume label 12-26
- Double quotation marks
  - HPFS 12-10
- Drive
  - alias 10-7
  - logical, EXTDSKDD.SYS
    - device driver 10-7
  - SET TEMP command 10-8
- DSR (data set ready) 12-57
- DTR (data terminal ready) 12-58
- Dynamic priority 9-10
- EAUTIL utility 13-9
- Environment variable
  - GROUPNAME 9-16
  - HOME 9-16
  - HOSTNAME 9-16
  - inheriting 9-16
  - SET KEYS 9-15
  - SET PATH 9-15
  - setting up 9-15
  - user, predefined 9-16
  - USERNAME 9-16
- Error
  - SYS0005 16-11
  - user maximum limit reached 16-9
- Error message
  - CONFIG.SYS 9-2
  - maximum number of users 12-5
- Esprit OPUS 3n1+ Terminal D-5
- Event category 14-4
- Event log flags
  - system profile 5-9
- Event Logging 1-12
- Event Manager 14-2
- Event tracing 9-24
- EXTDSKDD.SYS device driver 10-7
- Extended attributes
  - HPFS names, preserving 12-10
  - preserving 12-17
  - restoring 12-20
- Extended Partition
  - creating 3-9
  - deleting 3-15
- Extended partitions
  - creating, logical drives 3-10
- External disk drive 10-7
- External terminal 12-56
- FDISK utility 3-2
  - options 3-6
  - starting 3-5
- File
  - DISPLAY.DLL 10-5
  - copying HPFS to FAT 12-10
  - making backup copies 12-12
  - making backup, another
    - disk 12-13
  - moving HPFS to FAT 12-10
  - recovering, damaged FAT
    - disk 13-11



- restore, selecting 12-19
  - swapping 9-13
  - temporary 10-8
  - transferring HPFS to FAT 12-10
- File system
  - access 7-12
- File System Security 7-12
- Filename extension
  - .DDP 10-3
- First user program 4-2, 5-20
- Fixed disk
  - configuring 3-2
  - setting up 3-4
  - volume label 12-26
  - working with 12-21
- Fixed Disk Drive
  - selecting the next 3-18
- Fixed disk partition
  - cannot delete 16-8
- Floppy Disk
  - comparing 12-28
  - copying 12-25
  - formatting 12-22
  - volume label 12-26
  - working with 12-21
- Flow Control 6-21
- Format
  - 3½-Inch Disk 12-24
  - 5¼-Inch Disk 12-23
  - disk 12-21
  - floppy disk 12-22
  - log data 14-5
- FORMAT utility 12-21
- Font
  - printer 9-22
- Getting Help 2-7
- GLOSSARY Glossary-1
- GROUP PROFILES
  - Copying a Group 5-14
  - Deleting a Group 5-15
  - Modifying a Group 5-15
  - Renaming a Group 5-17
- GROUPNAME 1-3
  - environment variable 9-16
- Guest
  - security class 7-8
- Handshaking Techniques
  - connection type 6-23
  - data bits 6-20
  - data format 6-20
  - flow control 6-21
  - parity 6-21
  - stop bits 6-21
- HOME
  - environment variable 9-16
- Home directory 4-7
  - \USR\loginname 7-30
- Host Computer
  - defined 6-3
- Hotkeys
  - defined 6-3
  - list, Non-Shift Keys 6-17
  - list, Shift Keys 6-17
  - Login Switch 5-7, 6-16
  - Print Screen 5-7, 6-16
  - Print Screen Toggle 5-7, 6-16
  - Restart Terminal 5-7, 6-16, 9-27
  - Session Create 5-7, 6-16
  - Session Direct 5-7, 6-16
  - Session Switch 5-7, 6-16
  - settings, changing 6-31
  - Stop Program 5-7, 6-16
  - system console keyboard 9-27
  - System Dump 9-27
- HPFS 1-20
  - commands, using 12-10
  - copying files to FAT 12-10
  - moving files to FAT 12-10
  - naming features, case 12-10
  - naming features, characters 12-9
  - naming features,
    - commands 12-10
  - naming features, double quotation marks 12-10



- naming features, preserving in
    - extended attributes 12-10
  - naming files 12-9
  - transferring files to FAT 12-10
  - using files and directories 12-9
  - wildcard characters 12-11
- IBM 3151 Terminal
  - Models 51/61 D-8
- IFS 9-3
- Inch
  - lines per 12-55
- Infinite time-out 12-57
- Input handshake 12-57
- Input/output privilege levels 9-6
- Installation
  - Before you install 2-2
  - code page 9-18
  - CONFIG.SYS* 9-5
  - device drivers 10-3
  - file systems 2-3
  - installation disk 2-6
  - multiple fixed disks 2-6
  - multiple partitions 2-4
  - running install program 2-6
  - Security Attributes 7-28
  - starting 2-6
- IOPL command 9-6
  - enable 11-6
- Italic type 5
- Key
  - notational conventions 5
- Keyboard
  - DEVINFO 9-20
  - layout 9-21
- KEYBOARD.DCP 9-21
- Keynames 6
- KILL utility
  - terminate a process 12-51
- Kimtron KT-70PC Terminal D-10
- LABEL utility 12-27
- Language
  - code page 9-18
- Licensed
  - number of users, managing 12-5
- LIBPATH 9-3
- Line
  - characters per 12-55, 12-59
- Lines per inch 12-55
- Link MC2PC Terminal D-11
- Link MC5 Terminal D-13
- LOG 14-2
  - security, auditing 14-3
- Log file
  - formatting 14-5
  - naming 14-3
  - managing 14-3
- Logging Out 12-4
- Logical drive 3-3
  - creating 3-10
  - deleting 3-14
  - letter 10-7
  - volume label 12-26
- Login 1-4
  - administrator 12-2
  - example 12-3
  - handling, user 12-5
- Login procedure 12-2
- Login time restrictions 5-20
- Loginname 1-3
- Logout 12-4
  - administrator 12-7
  - examples 12-4, 12-7
- MAINTENANCE 9-3
- Maintenance mode 10-3, 13-2
  - enter, exit 13-3
- Managing memory 9-11
- Maximum Resource Limit 1-15, 5-4
- MAXWAIT command 9-3, 9-9
- MEMMAN command 9-3, 9-11
- Memory
  - BUFFERS command 9-14
  - DISKCACHE 9-11
  - MEMMAN command 9-12
  - SWAPPATH command 9-13
  - THREADS 9-8
  - virtual disk drive 10-8

- Minimum Resource Limit 1-15, 5-4
- MODE utility 12-54
  - asynchronous
    - communication 12-56
  - parallel printer 12-55
  - port, set up 10-6
  - serial port 12-56
  - setting display 12-59
- Modem 12-56
  - defined F-1
  - hardware, requirements 6-26
  - port, configuration F-3
  - setting up F-1
  - switch settings F-2
- Modifying
  - CONFIG.SYS* 9-5
- Monochrome display 12-59
- MSG utility 12-51
  - sending messages to users 12-51
- Multilingual code page 9-18
- Multitasking
  - MAXWAIT 9-9
  - modifying 9-7
  - PRIORITY 9-10
  - THREADS 9-8
  - TIMESLICE 9-10
- MultiTech 224E Modem F-3
- Multiuser
  - directory structure 4-7
- Naming
  - directories, FAT 12-9
  - directories, HPFS 12-9
  - files, FAT 12-9
  - files, HPFS 12-9
- Notational Conventions 5
- Okidata Okitel 2400 Modem F-3
- Operator
  - security class 7-7
- Option
  - notational conventions 5
- Output handshake 12-58
- Parallel
  - port 12-55
  - printer 9-22, 12-55
- Parity 6-21, 12-57
  - defined 6-3
- Partition 3-3
  - creating primary 3-7
  - deleting 3-13
  - displaying data 3-16
- PASSWORD 1-2, 7-4
  - assigning 12-5
  - Changing a User's Password 5-28
  - Days before warning 5-8
  - Maximum length 5-8
  - Maximum life 5-8
  - Minimum length 5-8
  - problem determination 16-6
  - Syntax 5-8
- Password settings
  - system profile 5-8
- PAUSEONERROR command 9-2
- Permissions 7-34
  - API 7-24
  - changing for multiple files 7-37
  - device 7-23
  - file system 7-22
  - Program Function Point
    - (PFP) 7-24
  - terminal 7-24
  - user function point 7-24
- PFPs
  - Program Function Points 7-17
- Plus sign (+) 6
- Port
  - asynchronous 12-56
  - communication, configure 8-6
  - MODE utility 12-55
  - parallel 12-55
  - printer, configure 8-8
  - serial arguments 12-56
  - serial, device driver 10-6
- Primary partition
  - deleting 3-16
- Print job
  - cancel 15-4
  - hold, job 15-6
  - print next 15-5

- release, job 15-6
- restart 15-5
- Print Queue
  - adding 8-2
  - cancel all jobs 15-4
  - cancel job 15-4
  - changing 8-5
  - deleting 8-5
  - hold 15-3
  - hold, job 15-6
  - print job next 15-5
  - release 15-4
  - release, job 15-6
  - restart job 15-5
- Print Spooler 7-14
- Printer
  - code page 9-18
  - DEVINFO 9-22
  - font table 9-22
  - parallel 12-55
  - port, configure 8-8
  - port, redirecting 8-12
  - security 7-14
  - serial 12-56
  - SPOOL utility 8-2
- Printing 1-17
  - configure a communication port 8-6
- PRIORITY command 9-10
  - ABSOLUTE 9-10
  - DYNAMIC 9-10
- Priority Considerations
  - Time Critical 11-11
- Privilege level
  - input and output 9-6
- Program Function Point
  - PFPs 7-17
  - SET\_TIME\_CRITICAL 11-12
- Program Selector 4-2
- PROTECTONLY 9-3
- QUERY utility 12-29
  - AUDIT 12-30
  - EVENTS 12-30
  - HOST 12-31
  - LIMITS 12-32
  - LIMITS, resource need 5-3
  - LOGIN 12-34
  - MEMORY 12-35
  - PRINT 12-38
  - PROCESS 12-38
  - SESSION 12-40
  - TERMINAL 12-42
  - USER 12-43
- README file
  - defined A-1
  - read, print, how to A-1
- REBOOT 9-3
- Receive Buffer Size 6-25
- RECOVER utility 13-4, 13-10, 13-11
- Refresh
  - configuration, parameters 6-33
  - menu, (F5) 7-35
- REGISTER utility 7-16
  - how to use 11-12
  - when to use 11-12
- Registered Programs 7-16
- RESET
  - AUDIT 12-48
  - EVENTS 12-49
  - LIMITS 12-49
  - TERMINAL 6-15, 12-50
- RESOURCE command 9-3, 9-26
  - =OFF, no limit check 5-3
- Resource limits 5-20
- Resource management 1-13
  - enabling and disabling 9-26
  - File handles 5-10
  - Logins 5-10
  - problem determination 16-8
  - Sessions 5-10
  - System semaphores 5-10
  - Threads 5-11
  - Virtual memory 5-10
- Restart Terminal Hotkey 9-27
- RESTORE utility 12-18, 12-20, 13-15
- Restoring
  - backup files 12-18
  - Extended Attributes 12-20

- files to a disk 12-18
- files, prompting user 12-19
- files, not on disk 12-20
- files, selecting 12-19
- Security Attributes 12-20
- subdirectories 12-18
- system 13-15
- ROM 9-22
- Rows per screen 12-59
- RS-232
  - wiring diagrams E-1
- RTS (request to send) 12-58
- RUN command 9-3, 9-17
- Screen
  - code page 9-18
  - device driver 10-4
  - DEVINFO 9-21
  - rows per 12-59
- Secondary Audit Events 7-28
- Security 1-9, 7-1
  - access checks, disabling 7-25
  - access checks, enabling,
    - disabling 9-26
  - access, fails 7-19
  - audit events 14-5
  - Audit Trail 14-2
  - audit, caution 14-6
  - auditing 7-26
  - classes 7-5
  - classes and file system
    - access 7-12
  - CONFIG.SYS* 11-5
  - considerations, directory
    - structure 11-2
  - error messages 16-11
  - load process 7-19
  - permissions 7-22
  - printer 7-14
  - problem determination 16-4
  - program loading 7-18
  - resource layout 7-29
- Security Attributes 7-9
  - Access Control List (ACL) 7-9
  - Audit Attributes 7-11
  - configuring 7-43
  - installation 7-28 Ownership 7-11
  - preserving 12-17
  - restoring 12-20
- Security classes
  - Administrator 7-6
  - Guest 7-8
  - Operator 7-7
  - User 7-8
- SECURITY command 9-3, 9-26
- Separator page file
  - creating 8-13
  - DEFAULT.SEP 8-4, 8-14
  - escape codes 8-13
- Serial Port
  - defined 6-3
  - reserving 6-34
- Session 1-4
- SET command 10-4
- SET COMSPEC 9-3
- SET DPATH 9-3
- SET KEYS 9-15
- SET PATH 9-3, 9-15
- SET PROMPT 9-3
- SET\_TIME\_CRITICAL
  - REGISTER, how to 11-10
  - REGISTER, when to 11-10
- SHUTDOWN 9-27, 12-8
- SPOOL utility 8-2
  - enable/disable 8-9
  - panel 15-2
  - panel, detail zoom 15-3
  - path, setting 8-10
  - printer 8-2
  - refresh rate 8-11
- Spooling 1-17
- Starting
  - program, background 9-17
  - system 2-7
- Startup file
  - batch files 4-7
  - CONFIG.SYS* 4-4
  - CONFIGUSR* 4-4
  - STARTUP.CMD* 4-4



- Startup options
  - terminal, different scenarios B-1
  - user, different scenarios B-1
  - STARTUP.CMD* 4-4
- Stop Bits 6-21, 12-57
  - defined 6-3
- Subdirectories
  - restoring 12-18
- SWAPPATH command 9-3, 9-13
- Swapping 9-13
- Switching
  - code pages 9-18
- Syntax
  - notational conventions 5
- SYS0005 16-11
- System
  - defaults, using 6-31
  - Logging 9-25, 14-3
  - shutting down 12-8
  - Profile 5-5
  - tracing 9-23
- System Console
  - defined 6-3
- System Dump Hotkey 9-27
- System Event Logging
  - (S)-Invalid Password 7-28
- System Global 11-9
  - problem determination 16-11
- System Global Shared Resources 1-16
- System profile
  - Audit flags 5-9
  - Default autologin user 5-6
  - Default first user program 5-6
  - Default hotkeys 5-7
  - Default resource limits 5-9
  - Event log flags 5-9
  - Password settings 5-8
  - Terminal timeout 5-6
- Televideo 965 Terminal D-16
- Temporary files 10-8
- Terminal 1-19
  - defined 6-3
  - external 12-56
  - problem determination 16-2
  - Security 7-14
  - setting up 6-9
  - startup options 1-21
  - termtype, defined D-2
- Terminal Configuration
  - parameters, resetting 6-32
- Terminal Profile
  - copying 6-29
  - creating, new 6-10
  - deleting 6-27
  - modify, Autologin Parameters 6-14
  - modify, Hotkeys 6-16
  - modify, Initial Terminal State 6-15
  - modify parameters 6-13
  - modify, Terminal Type 6-14
  - modifying 6-13
  - moving 6-28
  - renaming 6-27
- Terminal Subsystem
  - adding 6-6
  - adding, background 6-4
  - creating a new 6-33
  - defined 6-4
  - deleting 6-8
  - installing adapter 6-5
- Terminal timeout
  - system profile 5-6
- Terminal Type
  - changing 6-30
  - defaults, using 6-32
  - setting up specific D-1
- Terminals 1-19
- THREADS command 9-3, 9-8
- TIME
  - country settings 9-19
  - setting 12-52
  - setting, displaying 12-53
- Time Critical
  - priority considerations 16-12
  - REGISTER, how to 11-10
  - REGISTER, when to 11-10



- Time-out loop 12-55
- Time-out processing 12-57
- TIMESLICE command 9-10
- TRACE command 9-24TRACE Utility 9-25
- TRACEBUF command 9-24
- UFP
  - User Function Points 7-15
- United States code page 9-18
- User
  - identification 1-2
  - security class 7-8
  - state 5-19
- User function points 7-12, 7-15
- User Login Global 11-9
  - problem determination 16-11
- User Login Global Shared
  - Resources 1-16
- User Logins
  - handling 12-5
- User profile
  - Account Expiration 5-18
  - Audit flags 5-19
  - Auto connect login 5-19
  - Auto disconnect login 5-19
  - Class level 5-20
  - Default groupname 5-18
  - First user program 5-20
  - Login time restrictions 5-20
  - Resource limits 5-20
  - User state 5-19
  - Working directory 5-19
- USER PROFILES 5-18
  - Copying a User 5-22
  - Creating a New User 5-21
  - Deleting a User 5-23
  - Deleting a User From a Group 5-27
  - Modifying a User 5-24
  - Renaming a User 5-25
- User Program
  - setting up the first 4-2
- Username 1-3
- USERNAME
  - environment variable 9-16
- VDISK.SYS device driver 10-8
- Viewing
  - volume label 12-26
- VIOTBL.DCP 9-21
- Virtual disk drive
  - restarting system 10-8
  - VDISK.SYS device driver 10-8
- VOL command 12-26, 12-27
- Volume Label
  - assigning, changing 12-27
  - displaying 12-26
- Wildcard characters
  - in HPFS 12-11
- Working directory 4-7, 5-19
- Wyse 150 Terminal D-21
- Wyse 60 Terminal D-18





Citrix Systems, Inc.

210 University Drive, Suite 700, Coral Springs, FL 33071: Phone (305) 755-0559